

**UNIVERSIDAD CARLOS III DE MADRID**

**ESCUELA POLITÉCNICA SUPERIOR**

**INGENIERÍA DE TELECOMUNICACIÓN**



**PROYECTO FIN DE CARRERA**

***Interfaz para Teleoperación de un Robot Social  
desde Móvil***

**Autor: YOLANDA BUDIA ESCRIBANO**

**Tutor: IRENE PÉREZ ENCINAR**

**JUNIO DE 2015**



TÍTULO: *Interfaz para Teleoperación de un Robot Social desde Móvil*

AUTOR: *Yolanda Budia Escribano*

TUTOR: *Irene Pérez Encinar*

La defensa del presente Proyecto Fin de Carrera se realizó el día 25 de Junio de 2015; siendo calificada por el siguiente tribunal:

PRESIDENTE: *Álvaro Castro González*

SECRETARIO: *Esther Salichs San José*

VOCAL: *Lorena González Manzano*

Habiendo obtenido la siguiente calificación:

CALIFICACIÓN:

**Presidente**

**Secretario**

**Vocal**



# Agradecimientos

En primer lugar, agradecer a mi tutora Irene por su completa dedicación a lo largo de todo el proyecto. Me ha inspirado una gran motivación y ganas de querer mejorar el trabajo realizado.

A mi familia, y sobre todo a mis padres por su apoyo incondicional para todo y a mi hermano por ser mi guía, un ejemplo a seguir y por transmitirme sus inquietudes por aprender cosas nuevas e interesantes.

A mis amigas Sheila y Arancha por darme fuerzas cuando más lo necesitaba, por acompañarme en esta etapa de la vida siendo un gran apoyo para mí y por estar ahí en todo momento.

A mi pareja Jesús por aguantarme en este duro trayecto hasta el final haciendo que sea más fácil de superar.

Y a mis compañeros de la beca de Telemática por apoyarme, aconsejarme y ayudarme en la medida de lo posible.



*La manera de conseguir la felicidad  
es haciendo felices a los demás.*

- Robert Stephenson





# Resumen

La enfermedad de Alzheimer afecta cada vez a más personas a causa del envejecimiento de la población. Por esta razón, en este proyecto se quiere conseguir mejorar la calidad de vida de las personas que presentan este tipo de enfermedades, así como la de sus cuidadores.

Para el desarrollo de este trabajo se ha utilizado el robot social *Mini*, el cual ha sido desarrollado dentro del proyecto *RobAlz* por el grupo *RoboticsLab* del departamento de Sistemas y Automática de la *Universidad Carlos III de Madrid* (UC3M), en colaboración con la *Fundación Alzheimer España* (FAE). Este robot incorpora la arquitectura de software ROS (*Robot Operating System* – Sistema Operativo Robótico) y una base de datos MySQL en la que almacena la información necesaria de los escenarios de entretenimiento para los que está diseñado. Aunque *Mini* es un robot autónomo, se requiere de una interfaz gráfica de usuario que permita teleoperar dichos escenarios y así proporcionar tiempo libre al cuidador.

El objetivo principal del proyecto aquí presentado es diseñar, implementar y evaluar una interfaz gráfica para dispositivo móvil (*smartphone o tablet*), sencilla y fácil de usar para cualquier persona sin experiencia, que permita teleoperar las principales funcionalidades de entretenimiento del robot *Mini*, de manera remota e inalámbrica. Para ello la aplicación debe ser capaz de comunicarse tanto con la base de datos MySQL como con los nodos ROS del robot que incorpora.

En este trabajo se presentan los estudios realizados para llevar a cabo el diseño y la implementación de la aplicación y de los nodos ROS necesarios, para que el cuidador pueda consultar la actividad que está realizando el robot y seleccionar las tareas deseadas para que éste las realice.

**Palabras clave:** Alzheimer, robot social, *Mini*, ROS, MySQL, entretenimiento, interfaz gráfica de usuario, teleoperar, dispositivo móvil, *smartphone*, *tablet*, remota, inalámbrica, aplicación.



# Abstract

The Alzheimer's disease affects more and more people due to the population's ageing. Because of this fact, in this project it is wanted to achieve an improvement in the life quality of the people with this disease and of their carers.

The social robot *Mini* has been used for the evolution of this project. This robot has been developed in the *RobAlz* project by the *RoboticsLab* group that belongs to the Systems and Automatic department of the *University Carlos III of Madrid* (UC3M), in collaboration with the *Spain Alzheimer's Foundation* (FAE). *Mini* joins the ROS (Robot Operating System) software architecture as well as a MySQL database in which the information of the entertainment scenarios, for which it is designed, is stored. Even though *Mini* is an autonomous robot, it requires a graphical user interface that allows to teleoperate such scenarios and gives the carer free time.

The main objective of this project is to design, implement and evaluate a graphical interface for mobile devices (smartphone or tablet). This interface must be simple and easy to use for any inexperienced person. Moreover, it has to allow to teleoperate the main entertainment functionalities of the *Mini* robot, in a wireless and remote way. For this reason, the application must be able to communicate with the MySQL database as well as with the robot ROS nodes.

In this project, it is presented the studies carried out to design and implement the application and ROS nodes, which are necessary to allow the user checking the robot activity as well as choosing the desired tasks.

**Keywords:** Alzheimer, social robot, *Mini*, ROS, MySQL, entertainment, graphical user interface (GUI), teleoperate, mobile, smartphone, tablet, remote, wireless, application.



# Índice general

Introducción.....	25
1.1    Motivación del proyecto .....	26
1.2    Objetivos del proyecto .....	26
1.3    Estructura de la memoria .....	27
Planteamiento del problema .....	29
2.1    Enfermedad de Alzheimer (EA).....	30
2.2    Robot social <i>Mini</i> y enfermos de Alzheimer .....	31
2.3    Teleoperación .....	33
2.4    ROS ( <i>Robot Operating System</i> ) .....	34
2.4.1    Conceptos básicos .....	35
2.4.2    Bibliotecas de cliente .....	36
2.5    Dispositivos Móviles .....	36
Análisis .....	39
3.1    Diagrama de casos de uso .....	40
3.2    Casos de uso.....	41
3.2.1    Descripción de los atributos de los casos de uso .....	41
3.2.2    Casos de uso en formato extendido .....	41
3.3    Análisis de requisitos .....	45
3.3.1    Descripción de los atributos de los requisitos .....	45
3.3.2    Requisitos de información (RI) .....	46
3.3.3    Requisitos funcionales (RF) .....	47
3.3.4    Requisitos no funcionales (RNF) .....	49
Diseño.....	53
4.1    Diseño de las páginas de la aplicación.....	54
4.1.1    Página principal y menú .....	54
4.1.2    Páginas de las funcionalidades .....	54
4.1.3    Página de ayuda y página de resultados de búsqueda .....	55
4.2    Diagramas de flujo.....	56

---

4.2.1	Realizar tarea .....	56
4.2.2	Parar tarea .....	57
4.2.3	Pausar tarea.....	58
4.2.4	Reanudar tarea .....	59
4.2.5	Consultar estado del robot .....	59
4.3	Esquema del sistema .....	60
Implementación de la versión 1.0.....		61
5.1	Aplicación web (v.1.0).....	62
5.1.1	Tipos de aplicaciones móviles .....	62
5.1.2	Comparativa de los tipos de aplicaciones.....	66
5.1.3	<i>Frameworks</i> .....	67
5.1.4	Comparativa de los tipos de <i>frameworks</i> .....	71
5.2	Tecnologías utilizadas .....	72
5.2.1	Tecnologías de cliente .....	72
5.2.2	Tecnologías de servidor .....	76
5.2.3	Base de datos .....	77
5.3	Comunicación cliente – servidor .....	78
5.3.1	Aplicación – Base de datos.....	78
5.3.2	Aplicación – ROS.....	87
5.4	Diagramas de secuencia.....	92
5.4.1	Consultar ayuda de la aplicación .....	92
5.4.2	Cambiar idioma de la aplicación .....	92
5.4.3	Consultar estado del robot .....	93
5.4.4	Buscar tarea .....	93
5.4.5	Seleccionar funcionalidad.....	94
5.4.6	Seleccionar tarea.....	94
5.4.7	Realizar tarea .....	95
5.4.8	Pausar/reanudar tarea.....	95
5.4.9	Parar tarea .....	96
5.5	Estructura de ficheros aplicación v.1.0 .....	96
5.6	Capturas de la aplicación v.1.0 .....	98
5.6.1	Página principal y menú .....	99
5.6.2	Páginas de las funcionalidades .....	99
5.6.3	Página de ayuda y resultados de búsqueda.....	100
5.6.4	Cuadros de confirmación e información .....	102
Pruebas y evaluación .....		103
6.1	Pruebas.....	104

6.2	Evaluación .....	105
6.2.1	Perfil de usuario.....	105
6.2.2	Usabilidad de la interfaz .....	108
6.2.3	Funcionalidades .....	109
6.2.4	Valoración final.....	111
Mejoras y aplicaciones v.2.0 y v.2.1 .....		113
7.1	Aplicación web (v2.0).....	114
7.1.1	Diseño de la aplicación v.2.0 .....	114
7.1.2	Implementación de la aplicación v.2.0 .....	115
7.1.3	Mejoras en v.2.0.....	120
7.2	Aplicación híbrida (v2.1) .....	124
7.2.1	Implementación de la aplicación v.2.1 .....	124
7.2.2	Mejoras en v.2.1.....	125
7.3	Comparativa de versiones .....	126
7.3.1	Versión 1.0 .....	126
7.3.2	Versión 2.0 .....	126
7.3.3	Versión 2.1 .....	126
7.3.4	Comparación.....	127
Gestión del proyecto.....		129
8.1	Fases del desarrollo.....	130
8.2	Diagramas de Gantt .....	130
8.2.1	Planificación inicial .....	130
8.2.2	Planificación final.....	133
8.3	Presupuesto del proyecto .....	137
8.3.1	Horas dedicadas .....	137
8.3.2	Personal .....	137
8.3.3	Equipos .....	139
8.3.4	Otros costes .....	140
8.3.5	Presupuesto total.....	141
8.3.6	Plantilla resumen .....	141
Conclusiones y líneas futuras .....		147
9.1	Conclusiones .....	148
9.2	Líneas futuras.....	150
Encuesta.....		153
A.A	Perfil de usuario .....	154
A.B	Usabilidad de la interfaz .....	155
A.C	Actividad "Ver ayuda" .....	156

A.D	Actividad "Cambiar de idioma" .....	157
A.E	Actividad "Consultar el estado del robot" .....	157
A.F	Actividad "Ver lista de música" .....	158
A.G	Actividad "Ver lista de canales de Tv/Videos" .....	158
A.H	Actividad "Ver lista de historias" .....	159
A.I	Actividad "Ver lista de contenidos multimedia" .....	160
A.J	Realizar búsqueda .....	162
A.K	Experiencia general.....	162
	Manual de usuario .....	163
B.1	Acceder a la aplicación .....	164
B.2	Seleccionar idioma.....	165
B.3	Consultar estado del robot .....	166
B.4	Menú .....	166
B.5	Cambiar idioma.....	167
B.6	Buscar tarea.....	168
B.7	Funcionalidades .....	170
B.7.1	Música .....	170
B.7.2	Fotos .....	170
B.7.3	Tv/Videos.....	171
B.7.4	Noticias.....	172
B.7.5	Historias.....	172
B.7.6	Multimedia .....	173
B.8	Seleccionar tarea .....	174
B.8.1	Seleccionar canción .....	174
B.8.2	Seleccionar etiquetas .....	176
B.9	Consultar ayuda .....	179
B.10	Salir de la aplicación.....	179
B.11	Mensajes de error de conexión .....	180
B.11.1	Error de conexión con el servidor .....	180
B.11.2	Error de conexión con la base de datos .....	181
	Tecnologías auxiliares .....	183
C.1	PHPMyAdmin .....	184
C.2	Roslaunch: ficheros de lanzamiento .....	184
	Acrónimos .....	187
	Referencias .....	189



## Lista de Figuras

Figura 2.1 - Pirámide de edades .....	30
Figura 2.2 - Robot “ <i>Mini</i> ” .....	32
Figura 2.3 - Elementos del sistema de teleoperación .....	33
Figura 2.4 - Comunicación asíncrona entre nodos .....	35
Figura 2.5 - Comunicación síncrona entre nodos .....	36
Figura 2.6 - Sistemas Operativos Móviles.....	38
Figura 3.1 - Modelo de casos de uso del cuidador .....	40
Figura 4.1 - Diseño: página principal (izq.) y menú (dcha.) .....	54
Figura 4.2 - Diseño: página de música (izq.) y página de fotos (dcha.) .....	55
Figura 4.3 - Diseño: página de tv/videos (izq.) y página de noticias (dcha.) .....	55
Figura 4.4 - Diseño: página de historias (izq.) y página de multimedia (dcha.).....	55
Figura 4.5 - Diseño: página de ayuda (izq.) y página de búsqueda (dcha.).....	56
Figura 4.6 - Diagrama de flujo: Realizar tarea .....	57
Figura 4.7 - Diagramas de flujo: Parar tarea (izq.), Pausar tarea (medio) y Reanudar tarea (dcha.) .....	58
Figura 4.8 - Diagrama de flujo: Consultar el estado del robot .....	59
Figura 4.9 - Esquema general del sistema .....	60
Figura 5.1 - Plantillas gratuitas de Bootswatch .....	75
Figura 5.2 - Esquema general: Aplicación – Base de datos.....	78
Figura 5.3 - Esquema básico de una aplicación web .....	78
Figura 5.4 - Arquitectura de la aplicación web.....	79
Figura 5.5 - Esquema de comunicación: Aplicación – Base de datos .....	80
Figura 5.6 - Diagrama de secuencia: Aplicación – Base de datos .....	81
Figura 5.7 - Esquema de comunicación: Base de datos .....	81
Figura 5.8 - Relación entre las tablas y vistas MySQL de “robalz” .....	82
Figura 5.9 - Código creación vista “teleop_mobile” .....	85
Figura 5.10 - Código creación tabla “state_content” .....	86
Figura 5.11 - Esquema general: Aplicación – ROS .....	87
Figura 5.12 - Esquema de comunicación: Aplicación – ROS .....	88

Figura 5.13 - Diagrama de secuencia: Aplicación - ROS.....	88
Figura 5.14 - Esquema de comunicación: <i>Rosbridge</i> .....	89
Figura 5.15 - Esquema de comunicación: Nodos ROS .....	90
Figura 5.16 - Pseudocódigo del servicio “request_activity” .....	90
Figura 5.17 - Pseudocódigo del servicio “current_activity” .....	91
Figura 5.18 - Pseudocódigo del tema “activity_finished” .....	92
Figura 5.19 - Diagrama de secuencia: consultar ayuda de la aplicación.....	92
Figura 5.20 - Diagrama de secuencia: cambiar idioma de la aplicación .....	92
Figura 5.21 - Diagrama de secuencia: consultar estado del robot.....	93
Figura 5.22 - Diagrama de secuencia: buscar tarea .....	93
Figura 5.23 - Diagrama de secuencia: seleccionar funcionalidad .....	94
Figura 5.24 - Diagrama de secuencia: seleccionar tarea .....	94
Figura 5.25 - Diagrama de secuencia: realizar tarea .....	95
Figura 5.26 - Diagrama de secuencia: pausar /reanudar tarea.....	95
Figura 5.27 - Diagrama de secuencia: parar tarea .....	96
Figura 5.28 - Estructura de ficheros de la aplicación web v.1.0.....	97
Figura 5.29 - Web v.1.0: Página principal (izq.) y Menú desplegado (dcha.) .....	99
Figura 5.30 - Web v.1.0: Página de música (izq.) y Página de multimedia (dcha.).....	100
Figura 5.31 - Web v.1.0: Página de ayuda (izq.) e Insertar búsqueda (dcha.) .....	101
Figura 5.32 - Web v.1.0: Búsqueda de fotos (izq.) y Búsqueda general (dcha.) .....	101
Figura 5.33 - Web v.1.0: Búsqueda de fotos (izq.) y Búsqueda de multimedia (dcha.)	101
Figura 5.34 - Web v.1.0: Cuadros de Confirmación (izq.) e Información (dcha.).....	102
Figura 7.1 - Capas de la arquitectura cliente-servidor.....	114
Figura 7.2 - Comparativa entre aplicación web clásica y aplicación web con AJAX...	116
Figura 7.3 - Código de una función AJAX de ejemplo .....	117
Figura 7.4 - Estructura de ficheros del cliente.....	119
Figura 7.5 - Estructura de ficheros del servidor .....	120
Figura 7.6 - Mejoras v.2.0: Página inicial.....	121
Figura 7.7 - Mejoras v.2.0: Menú desplegado .....	121
Figura 7.8 - Mejoras v.2.0: Barra de búsqueda.....	122
Figura 7.9 - Mejoras v.2.0: Estado del robot .....	122
Figura 7.10 - Mejoras v.2.0: Botones funcionalidad multimedia .....	123
Figura 7.11 - Mejoras v.2.0: Cuadro información en multimedia .....	123
Figura 7.12 - Mejoras v.2.0: Identificación de botones .....	124
Figura 7.13 - Mejoras v.2.1: Acceso a la aplicación (izq.) y Salir (dcha.) .....	125
Figura 7.14 - Mejoras v.2.1: Mensaje error de conexión con el servidor (izq.) y con la base de datos (dcha.).....	126

---

Figura 8.1 - Esquema del ciclo de vida evolutivo .....	130
Figura 8.2 - Diagrama de Gantt: tareas de la planificación inicial .....	131
Figura 8.3 - Diagrama de Gantt de la planificación inicial .....	132
Figura 8.4 - Diagrama de Gantt: tareas planificación final .....	133
Figura 8.5 - Diagrama de Gantt de la planificación final .....	136
Figura 8.6 - Plantilla de presupuesto inicial .....	143
Figura 8.7 - Plantilla de presupuesto final .....	145
Figura B.1 - Acceso a la aplicación web e híbrida .....	164
Figura B.2 - Insertar URL en la aplicación web .....	165
Figura B.3 - Página de selección de idioma .....	165
Figura B.4 - Página principal/estado .....	166
Figura B.5 - Menú desplegado .....	167
Figura B.6 - Página principal/estado en otro idioma .....	168
Figura B.7 - Búsqueda de una tarea.....	169
Figura B.8 - Página de resultado de la búsqueda.....	169
Figura B.9 - Página de la funcionalidad “música” .....	170
Figura B.10 - Página de la funcionalidad “fotos” .....	171
Figura B.11 - Página de la funcionalidad “tv/videos” .....	171
Figura B.12 - Página de la funcionalidad “noticias” .....	172
Figura B.13 - Página de la funcionalidad “historias” .....	173
Figura B.14 - Página de la funcionalidad “multimedia” .....	173
Figura B.15 - Cuadro de confirmación .....	174
Figura B.16 - Cuadro de información.....	175
Figura B.17 - Actualización del estado.....	175
Figura B.18 - Cuadro de información al no seleccionar etiquetas.....	176
Figura B.19 - Selección de todas las etiquetas .....	177
Figura B.20 - Selección de algunas etiquetas .....	177
Figura B.21 - Cuadro de confirmación de las etiquetas seleccionadas .....	178
Figura B.22 - Cuadro de información de las etiquetas en reproducción .....	178
Figura B.23 - Página de la guía de ayuda .....	179
Figura B.24 - Cuadro de confirmación para salir de la aplicación .....	180
Figura B.25 - Error de conexión con el servidor .....	181
Figura B.26 - Error de conexión con la base de datos .....	181
Figura C.1 - Código activity_persistence.launch.....	186
Figura C.2 - Código para ejecutar activity_persistence.launch .....	186



## Lista de Tablas

Tabla 2.1 - Tipos de dispositivos móviles.....	37
Tabla 3.1 - CU-01: Cambiar idioma de la aplicación .....	41
Tabla 3.2 - CU-02: Consultar ayuda de la aplicación.....	42
Tabla 3.3 - CU-03: Consultar estado del robot .....	42
Tabla 3.4 - CU-04: Buscar tarea .....	42
Tabla 3.5 - CU-05: Seleccionar funcionalidad .....	43
Tabla 3.6 - CU-06: Seleccionar tarea.....	43
Tabla 3.7 - CU-07: Realizar tarea .....	44
Tabla 3.8 - CU-08: Parar tarea .....	44
Tabla 3.9 - CU-09: Pausar tarea.....	44
Tabla 3.10 - CU-10: Reanudar tarea .....	45
Tabla 3.11 - RI-01: Información sobre tareas disponibles.....	46
Tabla 3.12 - RI-02: Información sobre tareas seleccionadas.....	46
Tabla 3.13 - RF-01: Cambiar idioma de la aplicación.....	47
Tabla 3.14 - RF-02: Consultar ayuda de la aplicación .....	47
Tabla 3.15 - RF-03: Consultar estado del robot.....	47
Tabla 3.16 - RF-04: Buscar tarea.....	47
Tabla 3.17 - RF-05: Seleccionar funcionalidad .....	48
Tabla 3.18 - RF-06: Seleccionar tarea .....	48
Tabla 3.19 - RF-07: Realizar tarea.....	48
Tabla 3.20 - RF-08: Parar tarea .....	48
Tabla 3.21 - RF-09: Pausar tarea .....	48
Tabla 3.22 - RF-10: Reanudar tarea.....	49
Tabla 3.23 - RNF-01: Requisito de usabilidad .....	49
Tabla 3.24 - RNF-02: Requisito de facilidad de uso .....	49
Tabla 3.25 - RNF-03: Requisito de idioma.....	50
Tabla 3.26 - RNF-04: Requisito de tiempos de respuesta .....	50
Tabla 3.27 - RNF-05: Requisito de hardware.....	50

Tabla 3.28 - RNF-06: Requisito de conexión .....	50
Tabla 3.29 - RNF-07: Requisito de comunicación .....	51
Tabla 3.30 - RNF-08: Requisito de desempeño.....	51
Tabla 3.31 - RNF-09: Requisito de almacenamiento .....	51
Tabla 3.32 - RNF-10: Requisito de diseño e implementación.....	51
Tabla 3.33 - RNF-11: Requisito de la apariencia de la interfaz.....	52
Tabla 3.34 - RNF-12: Requisito de mantenibilidad.....	52
Tabla 3.35 - RNF-13: Requisito de estándares .....	52
Tabla 5.1 - Algunas características de SOs móviles .....	63
Tabla 5.2 - Ventajas e inconvenientes: Aplicaciones nativas.....	64
Tabla 5.3 - Ventajas e inconvenientes: Aplicaciones web .....	65
Tabla 5.4 - Ventajas e inconvenientes: Aplicaciones híbridas .....	65
Tabla 5.5 - Comparativa de los tipos de aplicaciones.....	66
Tabla 5.6 - Comparativa frameworks a) .....	71
Tabla 5.7 - Comparativa frameworks b).....	72
Tabla 5.8 - Tabla “multimedia_content” .....	83
Tabla 5.9 - Tabla “stories” .....	83
Tabla 5.10 - Vista “multimedia_tags” .....	84
Tabla 5.11 - Vista “tags” .....	84
Tabla 5.12 - Vista “teleop_mobile” .....	85
Tabla 5.13 - Matriz de equivalencia de tablas .....	85
Tabla 5.14 - Tabla “state_content” .....	86
Tabla 7.1 - Comparativa de las características de las versiones implementadas .....	127
Tabla 8.1 - Planificación inicial: Resumen de horas dedicadas.....	137
Tabla 8.2 - Planificación final: Resumen de horas dedicadas .....	137
Tabla 8.3 - Planificación inicial: Salario mensual de cada cargo .....	138
Tabla 8.4 - Planificación final: Salario mensual de cada cargo .....	138
Tabla 8.5 - Planificación inicial: Resumen de costes de personal .....	138
Tabla 8.6 - Planificación final: Resumen de costes de personal.....	139
Tabla 8.7 - Planificación inicial: Resumen de coste de equipos.....	139
Tabla 8.8 - Planificación final: Resumen de coste de equipos .....	140
Tabla 8.9 - Resumen de otros costes directos inicial .....	140
Tabla 8.10 - Resumen de otros costes directos final.....	140
Tabla 8.11 - Planificación inicial: Resumen del presupuesto sin I.V.A.....	141
Tabla 8.12 - Planificación final: Resumen del presupuesto sin I.V.A .....	141

## Lista de Gráficas

Gráfica 6.1 - A1. Rango de edad .....	106
Gráfica 6.2 - A2. Nivel de formación .....	106
Gráfica 6.3 - A3. Tipo de formación.....	107
Gráfica 6.4 - A4. Nivel de experiencia con <i>smartphones</i> .....	107
Gráfica 6.5 - A6. Conocimiento previo del robot .....	107
Gráfica 6.6 - Usabilidad de la interfaz.....	109
Gráfica 6.7 - Sencillez de acceso.....	109
Gráfica 6.8 - Funcionalidad intuitiva.....	111
Gráfica 6.9 - Funcionalidad sencilla.....	111
Gráfica 6.10 - K1. Expectativas iniciales .....	112





## Lista de Ecuaciones

Ecuación 8.1 - Cálculo del salario mensual.....	138
Ecuación 8.2 - Cálculo de la dedicación (hombre mes) .....	139
Ecuación 8.3 - Cálculo del salario mensual.....	139
Ecuación 8.4 - Cálculo de la amortización .....	140



# Capítulo 1

## Introducción

La robótica, como cualquier otra rama de la ciencia y de la tecnología, ha ido evolucionando con el tiempo y a su vez se han ido desarrollando las disciplinas que engloba, como: mecánica, electrónica, informática, inteligencia artificial, física, ingeniería de control, etc. Gracias a esta evolución, se ha fomentado el diseño e implementación de nuevos robots con el objetivo de mejorar la calidad de vida de la sociedad.

La mayoría de robots tienen como objetivo ayudar o incluso sustituir al ser humano en tareas que resultan ser tediosas, repetitivas, complicadas o peligrosas para éste. Inicialmente se utilizaba la robótica con fines industriales de manera que gran parte de los robots que existen en la actualidad son utilizados en líneas de producción. No obstante se está aumentando el desarrollo de robots de servicio capaces de realizar diversas tareas y que puedan estar presentes en diversos sectores de la sociedad como la agricultura, sanidad, educación, etc.

Actualmente, una de las áreas de la robótica que más importancia está cobrando en la actualidad es la conocida como *robótica social*, es decir la que se centra en el estudio de la integración de los robots en la sociedad humana. Dentro de los robots sociales, se encuentran los *robots sociales asistenciales*, cuyo objetivo es brindar asistencia a personas que necesitan supervisión, a través de la interacción. Dicha asistencia puede ofrecerse a diversos colectivos con necesidades específicas como por ejemplo: enfermos de Alzheimer, niños con autismo, pacientes con Parkinson, etc.

## 1.1 Motivación del proyecto

A causa del aumento del envejecimiento de la población, la enfermedad de Alzheimer (en adelante se denominará EA) afecta cada vez a más personas. Se trata de una enfermedad neurodegenerativa progresiva, irreversible y terminal, cuyo único tratamiento es básicamente mejorar, en la medida de lo posible, la calidad de vida de las personas con esta enfermedad y la de los cuidadores que las atienden.

Con la finalidad de asistir y entretener a personas que presentan esta enfermedad y también de ayudar a sus cuidadores, se ha construido el robot social *Mini* dentro del proyecto *RobAlz* en el que están trabajando el grupo de robótica social (*RoboticsLab*), del Departamento de Sistemas y Automática de la *Universidad Carlos III de Madrid* (UC3M), en colaboración con la *Fundación Alzheimer España* (FAE).

Aunque el robot *Mini* es básicamente autónomo, en este proyecto se plantea un caso de uso que requiere de una interfaz de teleoperación para que el cuidador de la persona con EA pueda disponer de tiempo libre. En dicho caso, el cuidador puede no estar presente físicamente con el enfermo de EA y el robot, sino que puede encontrarse en otra parte de la casa o incluso fuera de ésta de manera puntual, programando al robot para que entretenga al enfermo mediante la interfaz desde un dispositivo móvil.

## 1.2 Objetivos del proyecto

El objetivo principal del proyecto es diseñar, implementar y evaluar una interfaz gráfica para *smartphone*, sencilla y fácil de usar para cualquier persona sin experiencia, que permita teleoperar las principales funcionalidades de entretenimiento del robot *Mini*. Mediante dicha interfaz, se quiere conseguir aliviar la carga de los cuidadores de personas con EA, proporcionándoles tiempo libre cuando programen el robot para entretener al enfermo durante un periodo de tiempo.

Para lograr el objetivo principal, se ha realizado una subdivisión en un conjunto de objetivos más pequeños que debe cumplir la interfaz:

- Ser sencilla e intuitiva para usuarios de cualquier edad y nivel de conocimientos académicos y/o técnicos.
- Comunicarse con la arquitectura software del robot, es decir, con ROS (*Robot Operating System* – Sistema Operativo Robótico).
- Visualizarse en diferentes dispositivos y tamaños de pantalla.

- Utilizarse tanto en una red local como a través de Internet.
- Comunicarse con la base de datos del robot.
- Proporcionar información de la tarea que está realizando el robot.
- Permitir lanzar, pausar o parar tareas de entretenimiento del robot.

Con el fin de alcanzar estos objetivos, se han realizado los siguientes pasos a lo largo del proyecto:

- Estudio y familiarización con las diferentes tecnologías involucradas (aplicaciones móviles, ROS, bases de datos, etc.).
- Estudio del funcionamiento, características físicas, comportamientos básicos y funcionalidades existentes en el robot social *Mini*.
- Diseño e implementación de la interfaz.
- Pruebas unitarias, pruebas de usabilidad con un grupo de usuarios externo al desarrollo del proyecto, análisis de resultados e implementación de las sugerencias.

## 1.3 Estructura de la memoria

El presente documento se divide en un conjunto de capítulos que describen cada una de las distintas fases que componen el proceso de desarrollo del proyecto. A continuación se detalla la estructura del documento:

### Capítulo 1. Introducción

Es el primer capítulo del proyecto y en él se realiza una breve introducción y se presentan las motivaciones y objetivos principales para llevar a cabo el desarrollo del presente proyecto.

### Capítulo 2. Planteamiento del problema

En este capítulo se realiza un análisis del entorno tecnológico en el que se sitúa el proyecto. Comienza con una breve introducción a la EA y continúa con la descripción del robot social *Mini*. Después se detallan los elementos fundamentales de la teleoperación, se explican los conceptos más importantes de ROS y se realiza un estudio de los diversos dispositivos móviles así como la cuota de mercado de sus correspondientes sistemas operativos.

### Capítulo 3. Análisis

En la fase de análisis se especifican los requisitos y casos de uso que debe cumplir la aplicación a desarrollar.

**Capítulo 4. Diseño**

En esta etapa se realiza el diseño de cada una de las páginas que componen la aplicación así como de la información que se mostrará en cada una. También se describe la funcionalidad y el esquema de la aplicación en función de los requisitos y casos de uso definidos en el capítulo anterior.

**Capítulo 5. Implementación de la versión 1.0**

En esta fase se analizan las ventajas e inconvenientes de cada tipo de aplicación, se describen las tecnologías, herramientas y librerías utilizadas para su implementación, se describe cada bloque del esquema de la aplicación v.1.0 y los diagramas de secuencia teóricos correspondientes a cada caso de uso.

**Capítulo 6. Pruebas y evaluación**

En esta fase se presentan las pruebas realizadas para comprobar el correcto funcionamiento de la aplicación v.1.0 y se recogen los resultados obtenidos, tras poner a prueba el sistema y realizar una encuesta con un conjunto de usuarios ajenos al desarrollo del proyecto.

**Capítulo 7. Mejoras y aplicaciones v.2.0 y v.2.1**

Se detallan las mejoras incorporadas en las versiones 2.0 y 2.1 de la aplicación tras analizar los resultados de la evaluación con usuarios.

**Capítulo 8. Gestión del proyecto**

En este capítulo se detalla la planificación temporal inicial de las distintas fases que conforman el proyecto y el tiempo real que ha llevado su realización. Además se muestra el presupuesto en el cual se representan los costes asociados al proyecto para ambas planificaciones.

**Capítulo 9. Conclusiones y líneas futuras**

Finalmente se presentan las conclusiones obtenidas en la finalización del proyecto y las posibles mejoras a implementar en trabajos futuros.

**Anexos**

El documento incorpora tres anexos en los que se describen los contenidos necesarios para la completa comprensión del proyecto. En ellos se muestra la encuesta realizada a los usuarios en las pruebas, el manual de usuario con los pasos para utilizar la aplicación v.2.0 y v.2.1 y las tecnologías auxiliares utilizadas.

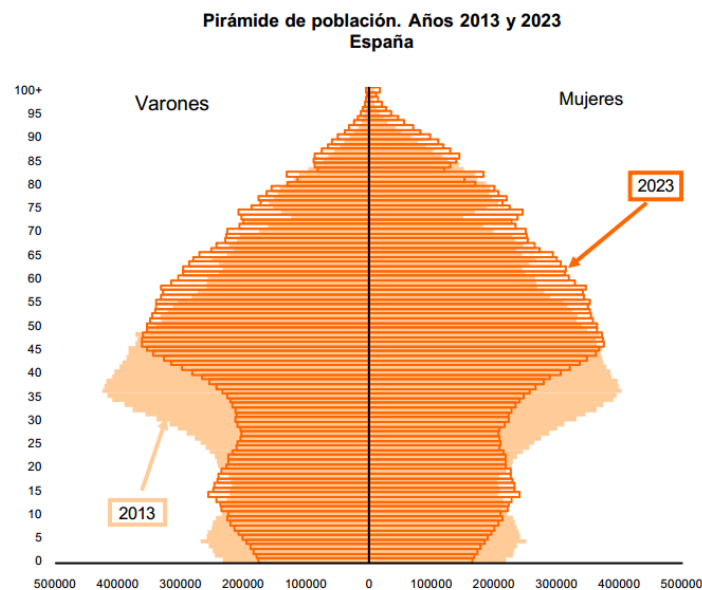
# Capítulo 2

## Planteamiento del problema

En este capítulo se realiza por un lado un estudio previo de la EA, sus fases y consecuencias y por otro lado, un estudio del robot *Mini* utilizado en el presente proyecto y un análisis de los distintos dispositivos y sistemas operativos móviles de la actualidad.

## 2.1 Enfermedad de Alzheimer (EA)

La disminución de la natalidad y el incremento de la esperanza de vida de la población a lo largo de los últimos años, han ocasionado el progresivo envejecimiento de la sociedad como se puede apreciar en la figura 2.1 (Fuente INe: Instituto Nacional de estadística [1]). A consecuencia de ello, ha aumentado la prevalencia de enfermedades neurodegenerativas como el Alzheimer.



*Figura 2.1 - Pirámide de edades*

Actualmente en España se observa una población envejecida y se estima que hay entre 500.000 y 800.000 pacientes con esta enfermedad [2] que aparece con mayor frecuencia en las personas mayores de 65 años de edad. La EA se trata de una patología neurodegenerativa progresiva, irreversible y terminal que produce pérdida de la memoria, cambios de personalidad y de humor, irritabilidad, dificultades motrices, confusión, desorientación temporal y espacial, etc., dependiendo de la etapa en la que se encuentre la enfermedad. Los enfermos de EA experimentan dificultades para el aprendizaje de nuevas tareas e incluso para realizar aquellas que ya tenían aprendidas como hablar, leer, comprender, escribir, vestirse, andar... También presentan problemas de coordinación y de planificación y es necesaria la supervisión y ayuda de un familiar o cuidador.

La enfermedad se divide en tres fases evolutivas [2, 3] aunque los límites entre cada fase no están claros y depende de cada persona. En ocasiones los enfermos de EA fallecen por otras causas y no pasan por todas las fases.

- 1ª fase (leve): las personas con EA comienzan a tener algunas pérdidas de memoria ya que olvidan hechos y datos recientes, nombres de objetos o de personas conocidas, fechas, y comienzan a desorientarse espacial y temporalmente. Pueden sufrir cambios de carácter y de personalidad, sentirse aislados y evitar gente y lugares nuevos.



- 2ª fase (moderada): se aumentan las deficiencias de la fase anterior convirtiendo al enfermo en una persona discapacitada. Disminuye la memoria reciente del enfermo, se desorientan en el tiempo y en el espacio, empiezan a perder el equilibrio y presentan dificultades en el lenguaje como repetir frases o no acabarlas, les cuesta relacionar y diferenciar objetos. En esta etapa el enfermo no podrá realizar por sí sólo tareas complejas por lo que es necesaria la supervisión de un familiar o cuidador que le ayude a realizar algunas tareas y prevenga que el enfermo se pierda o sufra accidentes.
- 3ª fase (grave): el paciente requiere de una atención constante por parte del cuidador o familiar. Aumenta la pérdida de la memoria tanto a corto como a largo plazo, pierden la capacidad de comunicarse, reconocer, comprender y controlar las funciones corporales, de modo que es incapaz de realizar tareas básicas como comer por sí mismo, asearse, vestirse, moverse, etc.

El tiempo de vida de las personas con EA varía de una a otra y en media pueden vivir entre 8 y 10 años desde que se diagnostica la enfermedad y ocasionalmente pueden llegar a los 20 años.

La dependencia de los enfermos de EA de terceras personas evoluciona con las etapas de la enfermedad. Los cuidadores se encargan de realizar las tareas complejas que el enfermo no es capaz de desempeñar por sí mismo y con frecuencia pueden afectar a su estado físico y emocional causándoles ansiedad y depresión. La dedicación de los cuidadores se torna exclusiva convirtiéndose así en “pacientes ocultos”.

Con el objetivo de mejorar la calidad de vida de los enfermos de EA y de sus cuidadores se han diseñado diversos robots sociales, como el que se presenta en el siguiente apartado.

## 2.2 Robot social *Mini* y enfermos de Alzheimer

El interés por la robótica ha ido creciendo en diversas áreas (minería, industria, hogar, etc.), siendo la robótica social una de las áreas de la robótica con más desarrollo en los últimos años.

El robot social es aquel que interactúa y se comunica con el ser humano siguiendo pautas y normas sociales. Dentro de éstos, se encuentran los *robots sociales asistenciales* que se dedican a la asistencia a través de la interacción. Algunos de estos robots se han aplicado a personas mayores o con algún tipo de demencia en diferentes áreas, como pueden ser la seguridad, la asistencia personal, el entretenimiento y la estimulación [4]. Por tanto, la apariencia de estos robots depende del tipo de aplicación que vaya a desempeñar.

En el proyecto *RobAlz* que está realizando el grupo de investigación de robótica, *RoboticsLab*, del Departamento de Sistemas y Automática de la UC3M en colaboración con la FAE, se ha diseñado el robot social asistencial *Mini* (figura 2.2). A diferencia de otros robots sociales aplicados a ancianos con demencia, en el diseño y desarrollo de *Mini* se ha tenido en cuenta las necesidades tanto del enfermo de EA como de sus cuidadores, para conseguir mejorar la calidad de vida de ambos.



*Figura 2.2 - Robot “Mini”*

### ▪ Características

El robot *Mini* es un robot estático, fácil de transportar, autónomo, que lleva incorporada una cámara *Kinect* que permite el reconocimiento facial de personas, una *tablet* para mostrar fotos, videos y más contenidos multimedia, un altavoz y micrófono para simular una conversación con el paciente, además de diversos sensores distribuidos en el cuerpo del robot para que al acariciarlo dé la sensación de que lo siente. Dispone de la plataforma Ubuntu 12.04 LTS (del inglés *Long Term Support*), utiliza el software ROS, su distribución *ROS Groovy Galapagos* y posee el servidor web *Apache*.

### ▪ Base de datos

Tiene implementada una base de datos MySQL denominada *robalz* que almacena los datos para personalizar escenarios y aplicaciones según los gustos del paciente.

### ▪ Escenarios

De los cuatro tipos de escenarios identificados en el “*Estudio de escenarios de uso para un robot social asistencial para enfermos de Alzheimer*” [4] (escenarios de seguridad, de asistencia personal, de entretenimiento y de estimulación), la interfaz a desarrollar en este proyecto para teleoperar el robot *Mini*, se va a centrar en los escenarios de entretenimiento.

Los escenarios de entretenimiento están compuestos por diversas funcionalidades cuya finalidad principal es entretener al paciente. En este caso, cabe la posibilidad de personalizar el comportamiento del robot en función de las preferencias del paciente con la ayuda del cuidador y/o familiares:

- *Cuenta-cuentos*: mediante esta funcionalidad, el cuidador puede seleccionar la historia, curiosidad, poema, hechos o eventos de la vida del paciente que desee que le cuente el robot.
- *Juegos*: el robot posee un repertorio de juegos de voz, como por ejemplo reconocer objetos, personas, etc., para entretener al enfermo durante un determinado periodo de tiempo.

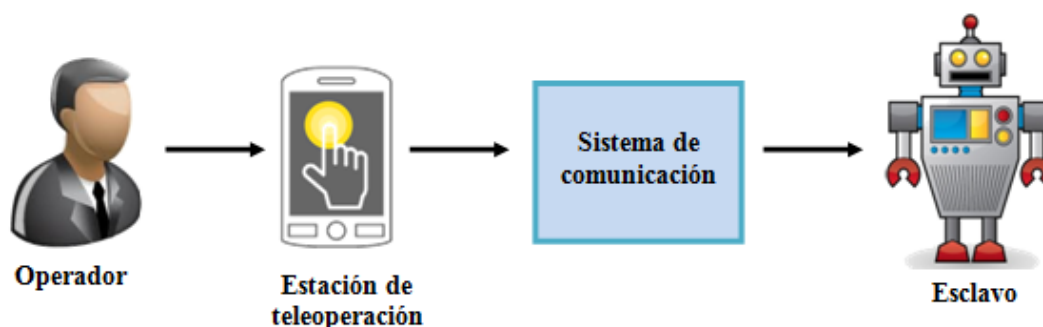
- *Contar noticias*: en función del interés del enfermo, el cuidador puede seleccionar las noticias de actualidad para que el robot se las cuente o se las muestre por pantalla en cualquier momento.
- *Reproductor multimedia*: con la finalidad de entretener al enfermo, el cuidador puede seleccionar diferentes contenidos multimedia de cualquier época: programas de televisión, de radio, canciones, acontecimientos históricos, fotos o vídeos caseros.

Aunque se trata de un robot autónomo, en este proyecto se requiere una **interfaz de teleoperación** para dispositivo móvil que permita al cuidador disponer de mayor tiempo libre, programando el robot sin tener que estar presente físicamente al lado del enfermo y del robot, desde otra habitación de la casa o desde fuera de ésta. Por tanto en los siguientes apartados de este capítulo se define el término teleoperación, se describen brevemente los conceptos básicos de ROS y se analiza la cuota de mercado de los dispositivos y SO móviles de la actualidad.

## 2.3 Teleoperación

Una de las áreas de la robótica es la telerrobótica que consiste en el control de un robot desde la distancia utilizando conexiones wireless (Wi-Fi, Bluetooth, y similares) o a través de Internet. Uno de los campos de esta área es la teleoperación que será una de las características más importantes de este proyecto. La teleoperación es el conjunto de tecnologías que permiten a un operario controlar un dispositivo a distancia.

El sistema de teleoperación es aquel que permite operar remotamente un dispositivo. Está compuesto principalmente por los siguientes elementos (figura 2.3): operador, estación de teleoperación, sistema de comunicación y esclavo. [5]



*Figura 2.3 - Elementos del sistema de teleoperación*

- El operador es el ser humano que controla de forma remota un dispositivo, denominado esclavo o dispositivo teleoperado, desde una estación de teleoperación por medio del sistema de comunicación.
- El esclavo es el dispositivo que actúa en el lugar remoto y es controlado por el operador. El esclavo puede ser un vehículo, un robot, un manipulador o cualquier dispositivo similar, pero en este proyecto el esclavo es el robot social que se encuentra al lado de la persona con deterioro cognitivo.

- La estación de teleoperación, que en este caso es el *smartphone*, está compuesta por un conjunto de entradas y salidas hardware y por un software, el cual tiene una interfaz gráfica de usuario que permite al operario interactuar con el robot a distancia.
- El sistema de comunicación es el que permite que se transmitan las señales de control al esclavo y se reciban las señales de información sobre el estado de éste en la estación de teleoperación. Las transmisiones y recepciones de estas señales se realizan mediante un canal de comunicación.

## 2.4 ROS (*Robot Operating System*)

ROS [6] es el *framework* que incorpora el robot *Mini* y en el que se basará todo el proyecto. Se trata de una arquitectura de código abierto utilizada por desarrolladores de software para la creación de aplicaciones robóticas. Es un conjunto de herramientas y bibliotecas que permite simplificar la creación del comportamiento de un robot en una amplia variedad de plataformas robóticas. Proporciona servicios de un Sistema Operativo (SO) incluyendo abstracción de hardware, control de dispositivos a bajo nivel, provisión de funcionalidades de uso común, mecanismo de paso de mensajes entre procesos y gestión de paquetes.

El principal objetivo de ROS es aumentar el desarrollo y la investigación de la robótica mediante la reutilización de código. Está compuesto por dos tipos básicos, el de SO y el de gestión de paquetes. Estos paquetes (del inglés *packages*) proporcionan la funcionalidad de ROS (por ejemplo, funciones de localización, escaneo, simulación, etc.), están organizados en carpetas denominadas pilas (del inglés *stacks*) y son desarrollados por los usuarios de la comunidad.

Actualmente, el software de ROS está orientado a plataformas basadas en Unix, tales como Linux Ubuntu (considerada la más estable y la principal plataforma de desarrollo) y Mac OS X, aunque la comunidad ofrece soporte a otras variantes consideradas en fase experimental como son Fedora, Gentoo, Arch Linux y otras plataformas Unix.

Es un *framework* distribuido de procesos en el que los ejecutables pueden ser diseñados en tiempo de ejecución de manera individual y se comparten y distribuyen de manera sencilla ya que están ordenados en paquetes y pilas. Así mismo, es de software libre bajo la licencia BSD (*Berkeley Source Distribution* - Distribución Fuente de Berkeley). Esta licencia se considera una licencia “permisiva” porque impone pocas restricciones sobre la forma de uso, modificaciones y redistribución del software.

Las distribuciones de ROS facilitan la instalación de una colección de software. Desempeñan un papel similar al de las distribuciones de Linux y también, mantienen versiones estables de un conjunto de software. Las distribuciones actuales son: *ROS Groovy Galapagos*, *ROS Hydro Medusa*, *ROS Indigo Igloo* y la próxima es *ROS Jade Turtle*.

### 2.4.1 Conceptos básicos

En este subapartado se van a definir los conceptos fundamentales de ROS: nodos, mensajes, temas y servicios.

ROS ha sido diseñado para ser modular, de manera que un sistema de control de un robot, generalmente estará compuesto por un conjunto de nodos. Los **nodos** (del inglés *nodes*) son procesos ejecutables que se encuentran en un paquete de ROS y se comunican entre sí por medio de paso de mensajes. Un **mensaje** (del inglés: *message*) es una estructura de datos que intercambian los nodos. Admite tipos primitivos estándar (entero, decimales, booleanos, etc.), así como arrays de tipos primitivos.

Un **tema** (del inglés: *topic*) es el medio por el cual los nodos se intercambian mensajes, mediante un sistema de publicación y subscripción. Se pueden diferenciar dos tipos de nodos: los subscriptores (del inglés: *subscribers*) son los nodos que se subscriben a un tema para recoger la información deseada y posteriormente tratarla, y los publicadores (del inglés: *publishers*) son los nodos que generan información y la publican en el tema correspondiente para que otros nodos puedan utilizarla.

Un nodo envía un mensaje mediante su publicación a un tema determinado. Un tema es un nombre que identifica el contenido del mensaje. El tipo del tema viene fijado por el tipo del mensaje publicado a dicho tema y los nodos sólo podrán recibir aquellos mensajes que se correspondan con el tipo determinado. Puede haber múltiples publicadores y subscriptores a un mismo tema y un nodo puede publicar y/o subscribirse a múltiples temas. Esta comunicación entre nodos es unidireccional (figura 2.4) de tipo muchos-a-muchos, de modo que no es apropiada para transacciones síncronas.

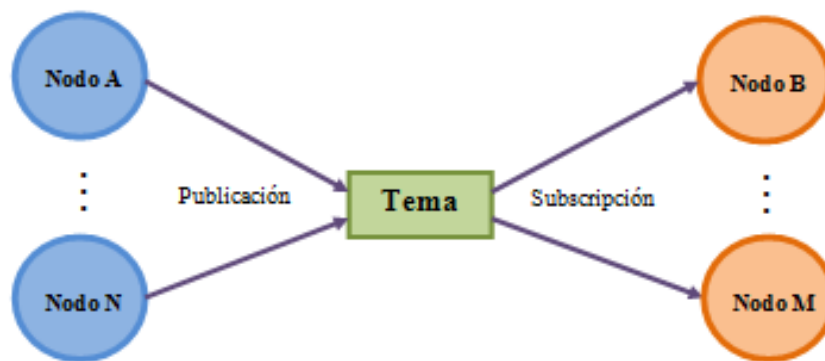


Figura 2.4 - Comunicación asíncrona entre nodos

Para el caso síncrono, es decir, para el caso en el que los nodos necesitan recibir una respuesta a una solicitud, se hace uso de una arquitectura cliente/servidor entre nodos (servicios). El **servicio** (del inglés: *service*) está definido por un par de mensajes, uno para la solicitud y otro para la respuesta. Como se muestra en la figura 2.5, un cliente llama a dicho servicio mediante el envío del mensaje de solicitud y espera una respuesta, a continuación el nodo servidor lo procesa y manda al cliente el mensaje de respuesta. El nodo servidor se mantiene a la espera de las solicitudes. De modo que esta comunicación entre nodos es bidireccional y bloqueante de tipo uno- a-uno.

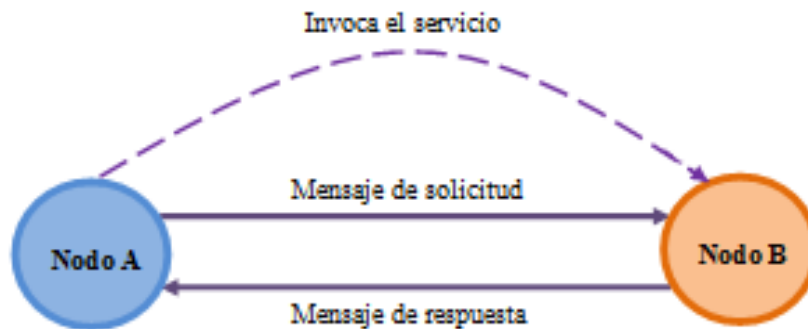


Figura 2.5 - Comunicación síncrona entre nodos

## 2.4.2 Bibliotecas de cliente

Las bibliotecas de cliente ROS tienen como objetivo facilitar el trabajo del programador. Se trata de un conjunto de códigos que permiten escribir nodos ROS, publicar y suscribirse a los temas, escribir y llamar a los servicios y usar el servidor de parámetros. Éste último es utilizado por los nodos para almacenar y recuperar parámetros en tiempo de ejecución.

Estas bibliotecas se pueden implementar en cualquier lenguaje de programación (C++, Python, Lisp, Java, Lua, Ruby, etc.), aunque se ofrece soporte para Python, y C++ está considerado como el más robusto. ROS dispone de una librería para cada lenguaje de programación como *roscpp* (lenguaje C++), *rospy* (lenguaje Python) y *roslisp* (lenguaje Common Lisp), sin embargo algunas se encuentran en fase experimental actualmente como *rosjava*, entre otras.

## 2.5 Dispositivos Móviles






Para lograr los objetivos del proyecto, es necesario analizar la cuota de mercado de cada dispositivo y SO móvil, previamente al diseño e implementación de la interfaz.

Los dispositivos móviles son pequeñas computadoras, conocidas también como computadoras de bolsillo o computadoras de mano (del inglés: *handheld*), que cuentan con las siguientes características: capacidades especiales de procesamiento, memoria limitada, conexión permanente o intermitente a una red y están diseñadas para una función específica pero pueden llevar a cabo funciones más generales.

Una de las principales características de estos dispositivos es el concepto de movilidad, debido a que su tamaño reducido permite que sean fácilmente empleados durante su transporte.

En la tabla 2.1 se muestran algunos tipos de dispositivos móviles que existen en la actualidad.



TIPOS DE DISPOSITIVOS MÓVILES		
Dispositivo móvil	Ejemplo	Descripción
<b>Teléfono inteligente (del inglés: <i>smartphone</i>)</b>		Teléfono móvil que dispone de un hardware y SO que permite realizar tareas y funciones extras a la realización y recepción de llamadas y mensajes telefónicos.
<b>Tableta (del inglés: <i>tablet</i>)</b>		Computadora portátil de tamaño superior al del <i>smartphone</i> pero inferior al del <i>netbook</i> . Se caracteriza por contar con una pantalla táctil de entre 7 y 12 pulgadas.
<b>Tabléfono (alternativa en español de <i>phablet</i>: contracción de <i>phone</i> y <i>tablet</i>)</b>		Dispositivos móviles con pantallas táctiles entre 5 y 7 pulgadas. Combina las funcionalidades y capacidades de un <i>smartphone</i> con una <i>tablet</i> .
<b>Reloj inteligente (del inglés: <i>smartwatch</i>)</b>		Es un reloj de pulsera que incorpora funciones mejoradas respecto a un reloj habitual. Incorporan tecnologías implementadas en <i>smartphones</i> , <i>tablets</i> y ordenadores, siendo capaces de ejecutar aplicaciones.
<b>Netbook</b>		Es un ordenador portátil de dimensiones reducidas y bajo coste. Proporciona una mayor movilidad y autonomía.

*Tabla 2.1 - Tipos de dispositivos móviles*

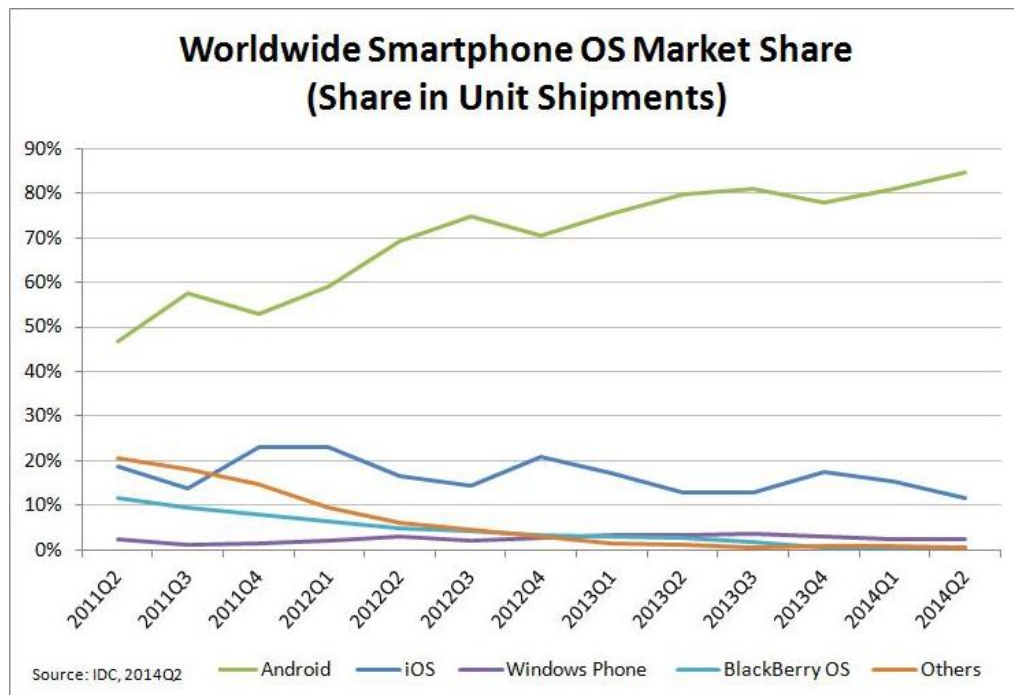
Hoy en día, la mayoría de la población dispone de uno o más dispositivos móviles, siendo el *smartphone* el más utilizado para acceder a Internet según desvela un estudio realizado por la AIMC (Asociación para la Investigación de Medios de Comunicación) [7]. Se prevé que este crecimiento continúe en los próximos años debido a que el *smartphone* permite al usuario una mayor movilidad y le proporciona una gran variedad de servicios en cualquier momento y lugar.

Por esta razón, uno de los requisitos del proyecto es que el dispositivo, en el que se va a utilizar la interfaz gráfica que teleoperará el robot, sea móvil (*smartphone* o *tablet*) debido a que este tipo de dispositivos permiten al cuidador teleoperar el robot desde una distancia remota a éste, de manera rápida y sencilla, permitiendo que pueda ausentarse de la estancia del enfermo a otra de la casa o incluso fuera de ésta.

Además de la gran variedad de dispositivos móviles que se encuentran en el mercado actualmente, hay disponibles distintos SOs para dichos dispositivos.

Este proyecto se centra principalmente en el uso del *smartphone* o *tablet* y por ello los SOs a estudiar son: Android, iOS, Windows Phone, BlackBerry OS, entre otros.

La IDC (*International Data Corporation* - Corporación de Datos Internacional) [8] ha realizado un análisis comparativo de los últimos años (figura 2.6). Según este análisis el SO operativo dominante es Android con casi el 85% de la cuota de mercado en el segundo cuatrimestre del 2014 (2014Q2), seguido de iOS con un 11,7%, en descenso en comparación con el segundo cuatrimestre del 2013(2013Q2).



*Figura 2.6 - Sistemas Operativos Móviles*



# Capítulo 3

## Análisis

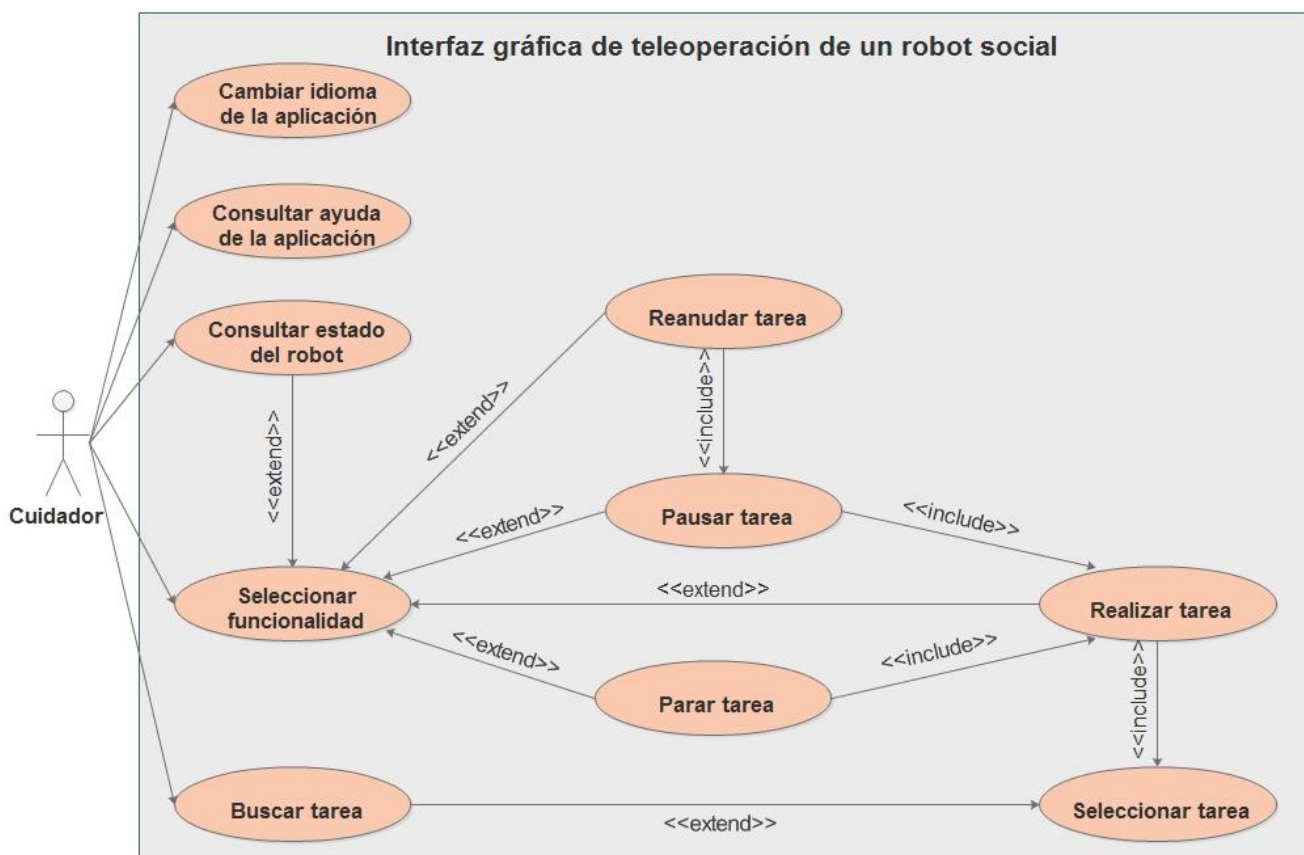
El desarrollo de la aplicación está orientado a teleoperar mediante una aplicación móvil, un robot social que ofrece diversos contenidos y funcionalidades de entretenimiento para el paciente. Por ello, este capítulo tiene como objetivo documentar las funcionalidades de la aplicación de modo sencillo. Está orientado a guiar y dirigir posteriormente el proceso de diseño e implementación del sistema propuesto.

En los siguientes subapartados se presentan los diagramas de casos de uso que describen las funcionalidades que ofrece el sistema de teleoperación y se realiza un análisis de los requisitos de información, funcionales y no funcionales que han sido obtenidos de los diferentes casos de uso.

En el apartado 2.2 se describen las funcionalidades correspondientes a los escenarios de entretenimiento en los que se va a centrar la aplicación a desarrollar. De modo que las principales funcionalidades que debe implementar la aplicación son: música, fotos, tv/videos, noticias, historias y multimedia. Debido a que el comportamiento de la aplicación es el mismo para todas, en los siguientes apartados de este capítulo, se denominarán las **funcionalidades de entretenimiento como “funcionalidades”** y las **canciones, fotos, canales de tv, videos, canales de noticias, historias y etiquetas como “tareas”**.

### 3.1 Diagrama de casos de uso

Para comprender el funcionamiento de la interfaz de teleoperación, en la figura 3.1 se muestra el diagrama de casos de uso de la aplicación a desarrollar. En total, el sistema está compuesto por diez funcionalidades que son realizadas por un actor, que en este caso es el cuidador de la persona con EA.



*Figura 3.1 - Modelo de casos de uso del cuidador*

## 3.2 Casos de uso

Los casos de uso son otra técnica para llevar a cabo la captura de requisitos de un sistema de software. En ellos se describen las secuencias de acciones entre el sistema y todos sus actores para conseguir un objetivo, pero de modo sencillo. Las descripciones se llevan a cabo evitando el empleo de tecnicismos y explicaciones acerca de la implementación del sistema, de modo que las funcionalidades quedan definidas de forma clara.

### 3.2.1 Descripción de los atributos de los casos de uso

En la definición de los casos de uso se consideran los siguientes aspectos:

- Nombre: se identifican con CU-XX y a continuación un nombre asociado.
- Actor: actores que inician el caso de uso
- Objetivo: breve descripción del objetivo del caso de uso
- Precondiciones: detallan las condiciones que se dan antes del caso de uso y su estado del sistema.
- Escenario básico: describe la secuencia de pasos en la interacción entre el usuario y el sistema de modo general, si el caso de uso finaliza correctamente.
- Postcondiciones: detallan las condiciones que se dan después del caso de uso y su estado del sistema.
- Escenario alternativo: describe los pasos alternativos si se produce algún error o restricción en el escenario básico.

### 3.2.2 Casos de uso en formato extendido

<b>CU-01</b>	<b>Cambiar idioma de la aplicación</b>
<b>Actor</b>	Cuidador.
<b>Objetivo</b>	Modificar el idioma de la aplicación al idioma seleccionado.
<b>Precondiciones</b>	-
<b>Escenario básico</b>	1. El usuario solicita a la aplicación cambiar el idioma. 2. La aplicación recarga la página con el idioma elegido.
<b>Postcondiciones</b>	El idioma de la aplicación se actualiza al idioma seleccionado.
<b>Escenario alternativo</b>	-

*Tabla 3.1 - CU-01: Cambiar idioma de la aplicación*

<b>CU-02</b>	<b>Consultar ayuda de la aplicación</b>
<b>Actor</b>	Cuidador.
<b>Objetivo</b>	Ver la ayuda de la aplicación para saber cómo interactuar con la interfaz para teleoperar el robot.
<b>Precondiciones</b>	-
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El usuario solicita a la aplicación consultar la guía de ayuda para utilizar la interfaz.</li> <li>2. La aplicación carga la página correspondiente a la ayuda.</li> <li>3. El usuario lee la guía de uso.</li> </ol>
<b>Postcondiciones</b>	El usuario ha visualizado correctamente la ayuda de la aplicación.
<b>Escenario alternativo</b>	-

*Tabla 3.2 - CU-02: Consultar ayuda de la aplicación*

<b>CU-03</b>	<b>Consultar estado del robot</b>
<b>Actor</b>	Cuidador.
<b>Objetivo</b>	Consultar el título y el tipo de tarea que está realizando el robot.
<b>Precondiciones</b>	-
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El usuario solicita a la aplicación consultar la tarea que está realizando el robot.</li> <li>2. La aplicación envía una petición al robot para obtener el nombre y tipo de la tarea que está realizando el robot.</li> <li>3. La aplicación recibe el nombre y tipo de tarea.</li> <li>4. La aplicación muestra el nombre y tipo de la tarea que está realizando el robot.</li> </ol>
<b>Postcondiciones</b>	Se muestra el título y el tipo de la tarea que está realizando el robot.
<b>Escenario alternativo</b>	-

*Tabla 3.3 - CU-03: Consultar estado del robot*

<b>CU-04</b>	<b>Buscar tarea</b>
<b>Autor</b>	Cuidador
<b>Objetivo</b>	Buscar un título o descripción de una tarea.
<b>Precondiciones</b>	-
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El usuario solicita a la aplicación buscar una tarea.</li> <li>2. La aplicación muestra una lista con las coincidencias encontradas.</li> </ol>
<b>Postcondiciones</b>	Se muestra una lista con los resultados de la búsqueda obtenidos.
<b>Escenario alternativo</b>	-

*Tabla 3.4 - CU-04: Buscar tarea*

<b>CU-05</b>	<b>Seleccionar funcionalidad</b>
<b>Actor</b>	Cuidador.
<b>Objetivo</b>	Ver la lista de tareas correspondientes a la funcionalidad seleccionada.
<b>Precondiciones</b>	-
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El usuario selecciona una funcionalidad en la aplicación.</li> <li>2. La aplicación muestra una lista con las tareas disponibles de la funcionalidad que ha seleccionado el usuario.</li> </ol>
<b>Postcondiciones</b>	Se muestra una lista con las tareas disponibles de la funcionalidad escogida por el usuario.
<b>Escenario alternativo</b>	-

*Tabla 3.5 - CU-05: Seleccionar funcionalidad*

<b>CU-06</b>	<b>Seleccionar tarea</b>
<b>Actor</b>	Cuidador
<b>Objetivo</b>	Seleccionar la tarea que va a realizar el robot.
<b>Precondiciones</b>	-
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. Se realiza el caso de uso CU05 – Seleccionar funcionalidad o CU04 – Buscar tarea.</li> <li>2. El usuario selecciona la tarea deseada de la lista.</li> <li>3. La aplicación solicita confirmación para enviar la tarea al robot.</li> </ol>
<b>Postcondiciones</b>	Se muestra un mensaje de confirmación.
<b>Escenario alternativo</b>	-

*Tabla 3.6 - CU-06: Seleccionar tarea*

<b>CU-07</b>	<b>Realizar tarea</b>
<b>Autor</b>	Cuidador.
<b>Objetivo</b>	Realizar la tarea seleccionada de la lista de una funcionalidad.
<b>Precondiciones</b>	-
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. Se realizan los casos de uso CU05 – Seleccionar funcionalidad o CU04 – Buscar tarea y CU06 – Seleccionar tarea.</li> <li>2. El usuario confirma la realización de la tarea.</li> <li>3. La aplicación envía un mensaje con la tarea seleccionada por el usuario en una petición al robot. Si el robot ya estaba realizando una tarea, ésta la finaliza para realizar la nueva tarea solicitada.</li> <li>4. El robot comienza a realizar la tarea solicitada.</li> <li>5. El sistema almacena la tarea que ha seleccionado el usuario y actualiza el estado del robot.</li> <li>6. La aplicación informa al usuario la tarea que está realizando el robot.</li> </ol>

<b>Postcondiciones</b>	El robot realiza la tarea que ha seleccionado el usuario, se actualiza el estado del robot y se muestra un mensaje de información.
<b>Escenario alternativo</b>	<ol style="list-style-type: none"> <li>2. El usuario cancela la realización de la tarea que ha seleccionado.</li> <li>2. La aplicación muestra de nuevo la lista de tareas disponibles.</li> <li>3. Si el robot ya está realizando una tarea y el usuario solicita la misma tarea, no comienza a realizarla de nuevo, continúa con la actual.</li> </ol>

*Tabla 3.7 - CU-07: Realizar tarea*

<b>CU-08</b>	<b>Parar tarea</b>
<b>Actor</b>	Cuidador.
<b>Objetivo</b>	Parar la tarea que está realizando el robot.
<b>Precondiciones</b>	El robot debe estar realizando una tarea.
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El usuario solicita a la aplicación parar la tarea que está realizando el robot.</li> <li>2. La aplicación envía un mensaje al robot.</li> <li>3. El robot para la tarea actual.</li> <li>4. El sistema elimina la tarea que se estaba realizando y actualiza el estado.</li> </ol>
<b>Postcondiciones</b>	El robot para la tarea y se actualiza el estado del robot.
<b>Escenario alternativo</b>	-

*Tabla 3.8 - CU-08: Parar tarea*

<b>CU-09</b>	<b>Pausar tarea</b>
<b>Actor</b>	Cuidador.
<b>Objetivo</b>	Pausar la tarea que está realizando el robot.
<b>Precondiciones</b>	El robot debe estar realizando una tarea correspondiente a las funcionalidades historias o multimedia.
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El usuario solicita a la aplicación pausar la tarea que está realizando el robot.</li> <li>2. La aplicación envía un mensaje al robot.</li> <li>3. El robot pausa la tarea actual.</li> </ol>
<b>Postcondiciones</b>	El robot pausa la tarea.
<b>Escenario alternativo</b>	-

*Tabla 3.9 - CU-09: Pausar tarea*

<b>CU-10</b>	<b>Reanudar tarea</b>
<b>Actor</b>	Cuidador.
<b>Objetivo</b>	Reanudar la tarea pausada.
<b>Precondiciones</b>	Debe haber una tarea pausada correspondiente a las funcionalidades historias o multimedia.
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El usuario solicita a la aplicación reanudar la tarea pausada.</li> <li>2. La aplicación envía un mensaje al robot.</li> <li>3. El robot continúa realizando la tarea pausada.</li> </ol>
<b>Postcondiciones</b>	El robot reanuda la tarea desde el instante en el que se pausó.
<b>Escenario alternativo</b>	-

*Tabla 3.10 - CU-10: Reanudar tarea*

### 3.3 Análisis de requisitos

Antes de iniciar el proceso de diseño de la aplicación, se realiza un análisis de requisitos en el que se definen las principales especificaciones y restricciones que debe cumplir. En dicho análisis se pueden diferenciar tres tipos de requisitos:

- Requisitos de información (RI): se encargan de indicar los datos que serán almacenados y consultados en el sistema.
- Requisitos funcionales (RF): son los requisitos que definen las funcionalidades e interacciones que tienen los usuarios con el sistema, es decir, detallan lo que debe hacer.
- Requisitos no funcionales (RNF): imponen criterios y restricciones en el diseño e implementación del sistema, es decir, describen cómo debe ser.

#### 3.3.1 Descripción de los atributos de los requisitos

Los campos definidos en los requisitos de información son los siguientes:

- Nombre: se identifican con RI-XX y a continuación un nombre asociado.
- Descripción: es una descripción breve del requisito.
- Datos asociados: detalla los datos relevantes que van a ser consultados o almacenados por la aplicación.

A continuación se describen los campos elegidos para la definición de los requisitos funcionales y no funcionales:

- Nombre: se identifican con RF-XX para los requisitos funcionales o RFN-XX para los requisitos no funcionales y a continuación un nombre asociado.
- Descripción: es una descripción breve del requisito.
- Prioridad: mide la importancia del requisito para la planificación en los procesos de diseño e implementación. Los valores que puede tomar son:

- Alta: el requisito debe ser implementado al inicio del desarrollo.
- Media: el requisito se debe implementar al finalizar los requisitos de prioridad alta.
- Baja: el requisito se debe implementar al finalizar los requisitos de prioridad media.
- Necesidad: determina si el requisito debe ser implementado de manera obligatoria u opcional. Puede tomar tres valores:
  - Esencial: el requisito tiene que ser implementado.
  - Deseable: no es obligatorio implementar el requisito pero sí es preferible.
  - Opcional: se puede implementar el requisito pero no es ni importante ni obligatorio.

### 3.3.2 Requisitos de información (RI)

Los requisitos de información especifican qué tipo de información relevante debe consultar y almacenar el software.

RI-01	Información sobre tareas disponibles
<b>Descripción</b>	El sistema deberá consultar la información correspondiente a las tareas del sistema. Dicha información depende del tipo de funcionalidad.
<b>Datos asociados</b>	<ul style="list-style-type: none"> <li>- Música: id, título, descripción, tipo y url.</li> <li>- Fotos: id, título, descripción, tipo y url.</li> <li>- Tv/Videos: id, título, descripción, tipo y url.</li> <li>- Noticias: id, título, descripción, tipo y url.</li> <li>- Historias: id, título, autor, tipo y referencia.</li> <li>- Multimedia: id, título, etiqueta y tipo.</li> </ul>

*Tabla 3.11 - RI-01: Información sobre tareas disponibles*

RI-02	Información sobre tareas seleccionadas
<b>Descripción</b>	El sistema deberá almacenar y consultar la información correspondiente a la tarea seleccionada por el usuario y que actualmente la está realizando el robot. Dicha información depende del tipo de funcionalidad.
<b>Datos asociados</b>	<ul style="list-style-type: none"> <li>- Música: tipo y url.</li> <li>- Fotos: tipo y url.</li> <li>- Tv/Videos: tipo y url.</li> <li>- Noticias: tipo y url.</li> <li>- Historias: tipo y referencia.</li> <li>- Multimedia: tipo y título.</li> </ul>

*Tabla 3.12 - RI-02: Información sobre tareas seleccionadas*



### 3.3.3 Requisitos funcionales (RF)

Los requisitos funcionales describen todas las funcionalidades del sistema. Definen por lo tanto el comportamiento interno del software a desarrollar y qué es lo que debe hacer con la información detallada en el apartado 3.3.2.

<b>RF-01</b>	<b>Cambiar idioma de la aplicación</b>		
<b>Descripción</b>	La aplicación debe permitir al usuario seleccionar el idioma al que desea cambiar la aplicación.		
<b>Prioridad</b>	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
<b>Necesidad</b>	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.13 - RF-01: Cambiar idioma de la aplicación*

<b>RF-02</b>	<b>Consultar ayuda de la aplicación</b>		
<b>Descripción</b>	La aplicación debe permitir al usuario consultar la guía que describe cómo utilizar la interfaz para teleoperar el robot.		
<b>Prioridad</b>	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
<b>Necesidad</b>	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.14 - RF-02: Consultar ayuda de la aplicación*

<b>RF-03</b>	<b>Consultar estado del robot</b>		
<b>Descripción</b>	La aplicación debe permitir al usuario consultar el título y el tipo de tarea que está realizando el robot en ese momento.		
<b>Prioridad</b>	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
<b>Necesidad</b>	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.15 - RF-03: Consultar estado del robot*

<b>RF-04</b>	<b>Buscar tarea</b>		
<b>Descripción</b>	La aplicación debe permitir al usuario buscar tareas. Es necesario que el usuario introduzca parte del título o de la descripción de la tarea que desea buscar. La aplicación mostrará una lista con las coincidencias que haya encontrado.		
<b>Prioridad</b>	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
<b>Necesidad</b>	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.16 - RF-04: Buscar tarea*

RF-05	Seleccionar funcionalidad		
Descripción	La aplicación debe permitir al usuario seleccionar una funcionalidad para ver su lista de tareas disponibles. Las funcionalidades que se deben incluir en la aplicación son: música, fotos, tv/videos, noticias, historias y multimedia.		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.17 - RF-05: Seleccionar funcionalidad*

RF-06	Seleccionar tarea		
Descripción	La aplicación debe permitir al usuario seleccionar una tarea de la lista correspondiente a las funcionalidades: música, fotos, tv/videos, noticias e historias. Y permitirá la selección múltiple para la funcionalidad multimedia.		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.18 - RF-06: Seleccionar tarea*

RF-07	Realizar tarea		
Descripción	La aplicación debe permitir al usuario confirmar la realización de la tarea que ha seleccionado.		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.19 - RF-07: Realizar tarea*

RF-08	Parar tarea		
Descripción	La aplicación debe permitir al usuario parar la tarea que está realizando el robot.		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.20 - RF-08: Parar tarea*

RF-09	Pausar tarea		
Descripción	La aplicación debe permitir al usuario pausar la tarea que está realizando el robot si las funcionalidades son historias y multimedia.		
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.21 - RF-09: Pausar tarea*

<b>RF-10</b>	<b>Reanudar tarea</b>		
<b>Descripción</b>	La aplicación debe permitir al usuario reanudar la tarea pausada si las funcionalidades son historias y multimedia.		
<b>Prioridad</b>	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
<b>Necesidad</b>	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.22 - RF-10: Reanudar tarea*

### 3.3.4 Requisitos no funcionales (RNF)

Los requisitos no funcionales detallan los criterios que debe seguir el sistema para realizar sus funcionalidades. Complementan a los requisitos funcionales especificando cualquier aspecto de su desarrollo.

- **Requisito de usabilidad**

Descripción de los requisitos que especifican la facilidad de uso del software para el usuario.

<b>RNF-01</b>	<b>Requisito de usabilidad</b>		
<b>Descripción</b>	La aplicación debe tener una interfaz gráfica sencilla para ser utilizada por cualquier tipo de usuario (de cualquier edad y conocimientos técnicos).		
<b>Prioridad</b>	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
<b>Necesidad</b>	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.23 - RNF-01: Requisito de usabilidad*

- **Requisito de facilidad de uso**

Especificación de requisitos que facilitan al usuario el empleo del software desarrollado.

<b>RNF-02</b>	<b>Requisito de facilidad de uso</b>		
<b>Descripción</b>	La aplicación debe constar de un manual que describa el funcionamiento y el uso del sistema al usuario final.		
<b>Prioridad</b>	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
<b>Necesidad</b>	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.24 - RNF-02: Requisito de facilidad de uso*

RNF-03	Requisito de idioma		
Descripción	La aplicación debe ser multilinguaje. Como mínimo debe disponer de dos idiomas: inglés y español.		
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.25 - RNF-03: Requisito de idioma*

- **Rendimiento**

Especificación de los requisitos relacionados con la carga que tiene que soportar el sistema.

RNF-04	Requisito de tiempos de respuesta		
Descripción	Los tiempos de respuesta del sistema deben ser de un máximo de cinco segundos.		
Prioridad	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Necesidad	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.26 - RNF-04: Requisito de tiempos de respuesta*

- **Requisito de hardware**

Requisitos que especifican el hardware necesario para que el software funcione.

RNF-05	Requisito de hardware		
Descripción	La aplicación debe ser soportada por dispositivos móviles de tipo <i>smartphone</i> o <i>tablet</i> .		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.27 - RNF-05: Requisito de hardware*

- **Requisito de comunicaciones**

Definen las características que debe tener el sistema para establecer la comunicación con el usuario.

RNF-06	Requisito de conexión		
Descripción	La aplicación hará uso de conexión a Internet, por lo que el dispositivo móvil debe disponer de conexión a Internet mediante WiFi o mediante los datos móviles que tenga contratado el usuario.		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.28 - RNF-06: Requisito de conexión*

RNF-07	Requisito de comunicación		
Descripción	La aplicación debe comunicarse con el robot mediante ROS de distribución <i>Groovy</i> .		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.29 - RNF-07: Requisito de comunicación*

- **Requisito de desempeño**

Especificación de requisitos que proporcionan las características de desempeño que debe tener el software.

RNF-08	Requisito de desempeño		
Descripción	El sistema no debe permitir la ejecución simultánea de tareas seleccionadas por el usuario.		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.30 - RNF-08: Requisito de desempeño*

- **Requisito de almacenamiento**

Determinan el tipo de almacenamiento y la ubicación de los datos.

RNF-09	Requisito de almacenamiento		
Descripción	Los datos del sistema deberán ser consultados y almacenados en una base de datos MySQL.		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.31 - RNF-09: Requisito de almacenamiento*

- **Requisito de diseño e implementación**

Requisitos que especifican las restricciones que debe cumplir el software para su diseño e implementación.

RNF-10	Requisito de diseño e implementación		
Descripción	La aplicación se debe adaptar a cualquier tamaño de pantalla y a los cambios de orientación (vertical y horizontal).		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Necesidad	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.32 - RNF-10: Requisito de diseño e implementación*

- **Requisito de la apariencia de la interfaz**

Descripción de los requisitos de la interfaz de usuario para el producto (estilo de pantallas, colores, etc.).

RNF-11	Requisito de la apariencia de la interfaz		
<b>Descripción</b>	El interfaz del sistema debe incluir pantallas, menús y submenús. Los diseños de las pantallas y el tamaño y tipo de letra deben de ser uniformes en toda la aplicación.		
<b>Prioridad</b>	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
<b>Necesidad</b>	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.33 - RNF-11: Requisito de la apariencia de la interfaz*

- **Restricción de mantenibilidad**

Definen los aspectos necesarios que permiten ampliar las funcionalidades y modificar o corregir errores de un sistema software.

RNF-12	Requisito de mantenibilidad		
<b>Descripción</b>	El código fuente del software debe estar debidamente documentado.		
<b>Prioridad</b>	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
<b>Necesidad</b>	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.34 - RNF-12: Requisito de mantenibilidad*

- **Requisito de estándares**

Describe las licencias de utilización que utiliza el software.

RNF-13	Requisito de estándares		
<b>Descripción</b>	La licencia de uso del software con el que se realiza la aplicación debe ser lo menos restrictiva posible, preferiblemente software de código abierto.		
<b>Prioridad</b>	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
<b>Necesidad</b>	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional

*Tabla 3.35 - RNF-13: Requisito de estándares*

# Capítulo 4

## Diseño

En este capítulo se muestra la apariencia que debe tener la aplicación y se realiza una descripción de su funcionamiento, así como del esquema general del sistema, teniendo en cuenta el análisis de requisitos y la especificación de los casos de uso del sistema, descritos en el capítulo 3.

## 4.1 Diseño de las páginas de la aplicación

En este apartado se muestra el diseño de cada una de las páginas de la aplicación. Está formada por las siguientes: página principal, una página por funcionalidad a implementar, página de ayuda, página de resultados de búsqueda y el menú desplegable accesible desde todas las páginas anteriores.

### 4.1.1 Página principal y menú

En la figura 4.1 se muestra el diseño de la página principal (imagen izquierda) y del contenido del menú (imagen derecha). La página principal contiene un botón por cada funcionalidad a implementar (música, fotos, tv/videos, noticias, historias y multimedia) y muestra el estado del robot (RF-03: Consultar estado del robot). El menú es común a todas las páginas que componen la aplicación. Desde éste se puede acceder a la página principal, a las páginas de cada funcionalidad (RF-05: Seleccionar funcionalidad) y a la página de ayuda (RF-02: Consultar ayuda). Además permite realizar búsqueda de tareas (RF-04: Buscar tarea) y cambiar el idioma de la aplicación (RF-01: Cambiar idioma de la aplicación).

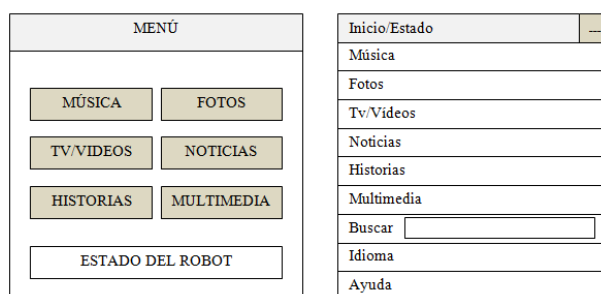


Figura 4.1 - Diseño: página principal (izq.) y menú (dcha.)

### 4.1.2 Páginas de las funcionalidades

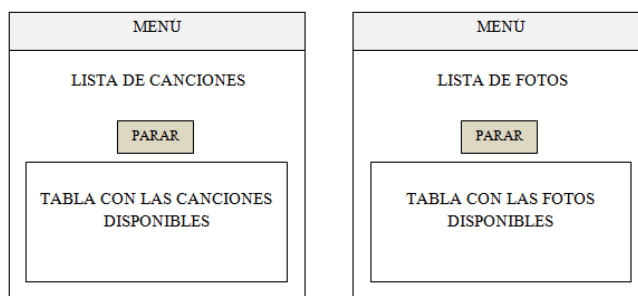
La aplicación tiene una página para cada funcionalidad: música (imagen izquierda de la figura 4.2), fotos (imagen derecha de la figura 4.2), tv/videos (imagen izquierda de la figura 4.3), noticias (imagen derecha de la figura 4.3), historias (imagen izquierda de la figura 4.4), y multimedia (imagen derecha de la figura 4.4).

Debido a la implementación ROS que posee el robot *Mini*, las funcionalidades música, fotos, tv/videos y noticias únicamente tienen la posibilidad de parar la tarea. Mientras que para las funcionalidades historias y multimedia, además de parar, se puede pausar y reanudar la tarea. Por esta razón, no hay el mismo número de botones en todas las páginas para dichas funciones.

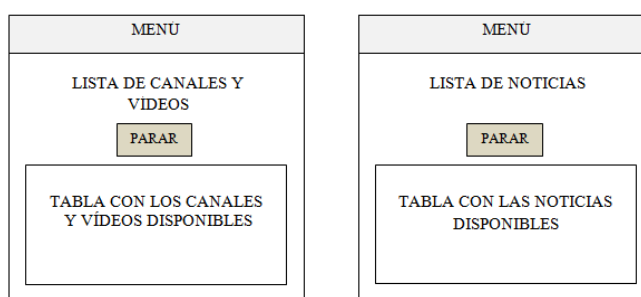
La habilidad multimedia permite seleccionar un conjunto de etiquetas para que el robot realice el contenido relacionado con ellas. La etiqueta sirve para asignar con un nombre un conjunto de contenidos multimedia que tienen algo en común. Por ejemplo, si el cuidador quiere que el robot le enseñe al enfermo fotos relacionadas con su nieto, seleccionaría la etiqueta *nieto* de la lista y el robot mostraría todas las fotos en las que aparece su nieto. Además, si se quiere mostrar específicamente las fotos en las que aparezca su nieto en la playa,



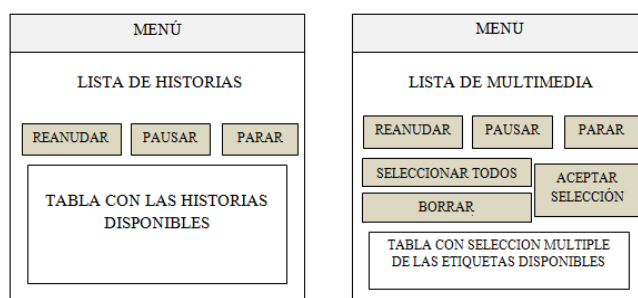
se necesita poder seleccionar ambas etiquetas, *nieto* y *playa*. De este modo, en la página de la funcionalidad multimedia se incorpora la opción de selección múltiple de etiquetas, por lo que se añade un botón para seleccionar todas, otro para borrar la selección y otro para aceptar la selección realizada.



*Figura 4.2 - Diseño: página de música (izq.) y página de fotos (dcha.)*



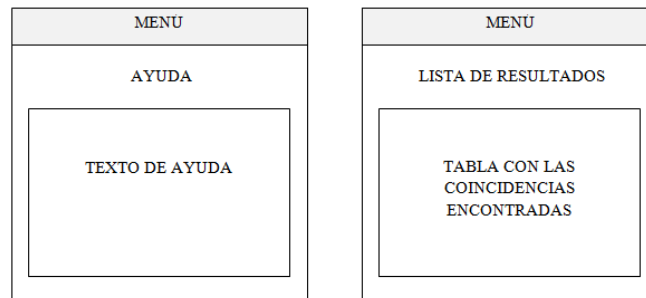
*Figura 4.3 - Diseño: página de tv/videos (izq.) y página de noticias (dcha.)*



*Figura 4.4 - Diseño: página de historias (izq.) y página de multimedia (dcha.)*

### 4.1.3 Página de ayuda y página de resultados de búsqueda

En la figura 4.5 se muestran los diseños de las páginas de ayuda (imagen de la izquierda) y de resultados de búsqueda (imagen de la derecha). La página de ayuda contiene la información necesaria para que el usuario sea capaz de utilizar la aplicación correcta y completamente. Y la página de resultados de búsqueda muestra las coincidencias encontradas tras introducir el texto de la tarea a buscar.



*Figura 4.5 - Diseño: página de ayuda (izq.) y página de búsqueda (dcha.)*

## 4.2 Diagramas de flujo

En base a los requisitos y casos de uso definidos, en los diagramas de flujo de la figura 4.6, figura 4.7 y figura 4.8, se representa gráficamente la secuencia básica de los eventos que se generan en la utilización de la aplicación al reproducir, parar, pausar y reanudar una tarea y al consultar el estado del robot. Cada uno de los diagramas se explica con un ejemplo.

### 4.2.1 Realizar tarea

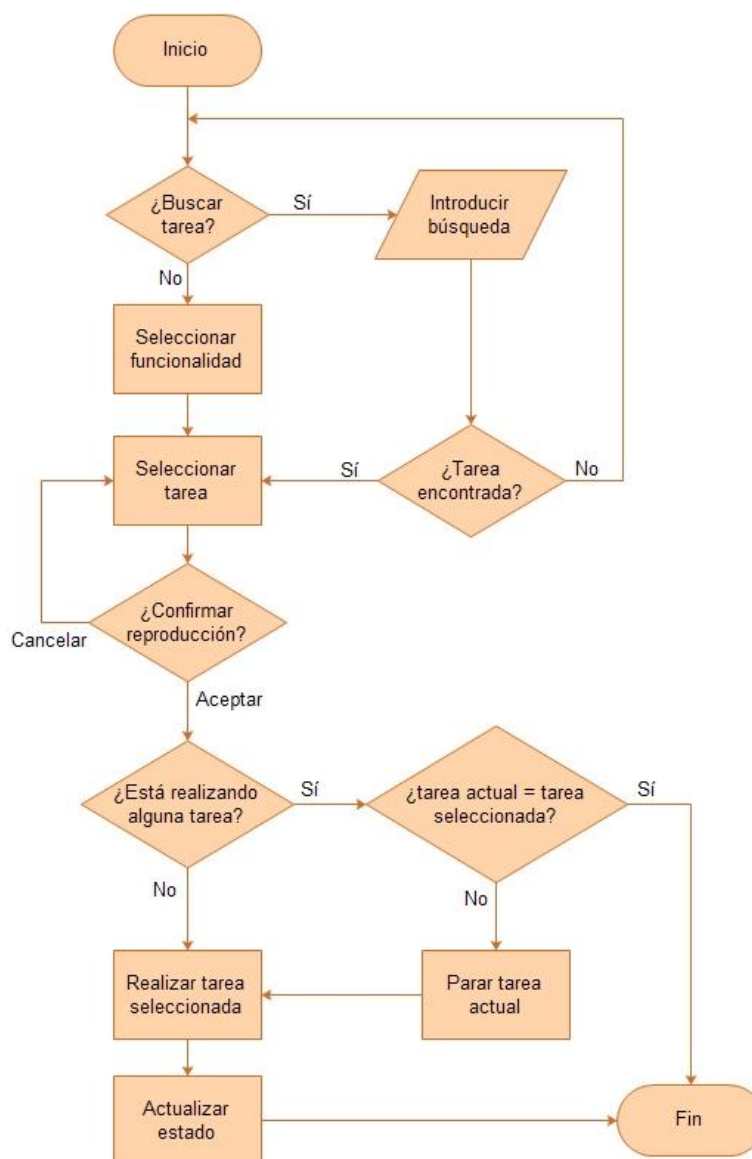
Cumpliendo con los requisitos RF-04 (Buscar tarea), RF-05 (Seleccionar funcionalidad), RF-06 (Seleccionar tarea) y RF-07 (Realizar tarea), en la figura 4.6 se muestra el diagrama de flujo al realizar una tarea.

Seguidamente se describe como ejemplo, el flujo resultante cuando el usuario (cuidador) quiere que el robot reproduzca una canción:

- 1- El usuario selecciona la funcionalidad música desde la página principal o desde el menú de la aplicación.
- 2- En la aplicación se muestra la página correspondiente con la lista de canciones.
- 3- El usuario escoge una de las canciones de la lista disponible.
- 4- La aplicación pregunta al usuario si está seguro de querer reproducir dicha canción.
- 5- El usuario acepta la reproducción de la canción en el cuadro de confirmación que aparece.
- 6- El robot comienza a reproducir la canción y actualiza su estado con la canción seleccionada.
- 7- La aplicación muestra un cuadro de información de la canción que está reproduciendo el robot.
- 8- La aplicación vuelve a la página de la lista de canciones.

Cumpliendo con el requisito RNF-08 (Requisito de desempeño), en el caso en el que el robot esté realizando alguna tarea, se comprueba si la seleccionada por el usuario es la misma que la que está realizando. Si es la misma el robot continúa con la tarea que estaba realizando, y si no lo es, el robot para la tarea que estaba realizando y comienza a realizar la nueva que ha seleccionado.

Además, el usuario puede buscar la canción o cualquier otra tarea introduciendo parte del título o descripción de ésta en la barra de búsqueda del menú desplegable, y la aplicación mostrará al usuario la página de resultados de búsqueda con las coincidencias encontradas en caso de que las haya.



*Figura 4.6 - Diagrama de flujo: Realizar tarea*

### 4.2.2 Parar tarea

Cumpliendo con el requisito RF-08 (Parar tarea), en el diagrama de la izquierda de la figura 4.7 se muestra el diagrama de flujo al parar una tarea.

Se describe como ejemplo, el flujo resultante cuando el usuario pulsa el botón “parar” para que el robot pare un video:

- 1- El usuario selecciona la funcionalidad tv/videos desde la página principal o desde el menú de la aplicación.

- 2- En la aplicación se muestra la página correspondiente con la lista de canales y videos.
- 3- El usuario pulsa el botón “parar” para finalizar la reproducción del video.
- 4- El robot para de reproducir el video y actualiza su estado.
- 5- La aplicación vuelve a la página de la lista de canales y videos.

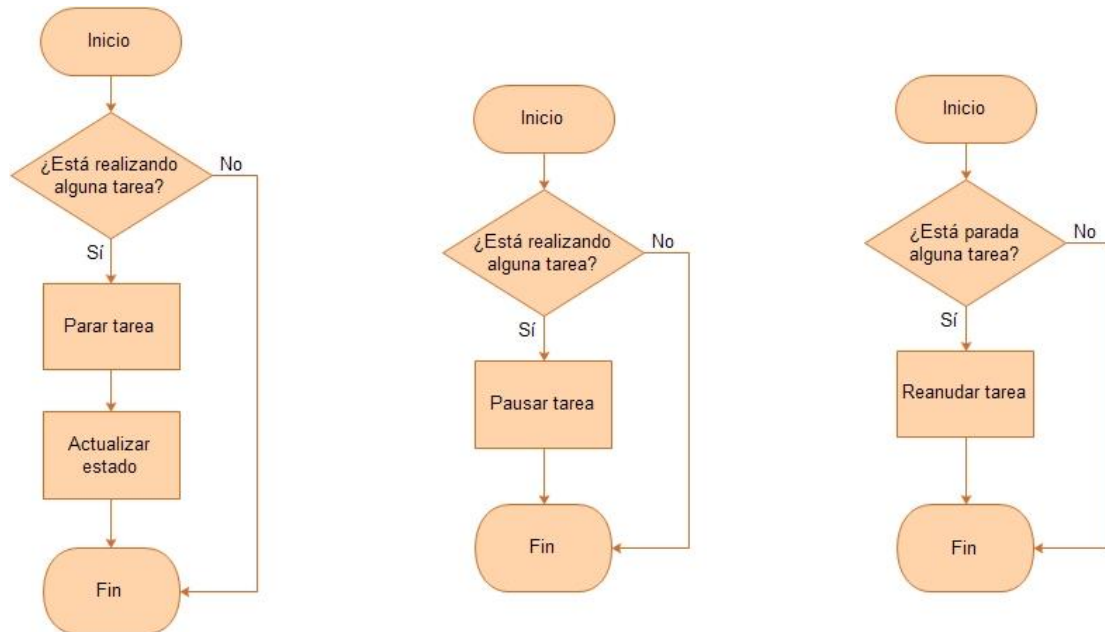


Figura 4.7 - Diagramas de flujo: Parar tarea (izq.), Pausar tarea (medio) y Reanudar tarea (dcha.)

### 4.2.3 Pausar tarea

Cumpliendo con el requisito RF-09 (Pausar tarea), en el diagrama del medio de la figura 4.7 se muestra el diagrama de flujo al pausar una tarea. Actualmente, sólo se pueden pausar las tareas de las funcionalidades historias y multimedia.

A continuación se detalla el flujo resultante cuando el usuario pulsa el botón “pausar” para que el robot pause una historia:

- 1- El usuario selecciona la funcionalidad historias desde la página principal o desde el menú de la aplicación.
- 2- En la aplicación se muestra la página correspondiente con la lista de historias.
- 3- El usuario pulsa el botón “pausar” para pausar la historia.
- 4- El robot hace una pausa.
- 5- La aplicación vuelve a la página de la lista de historias.

#### 4.2.4 Reanudar tarea

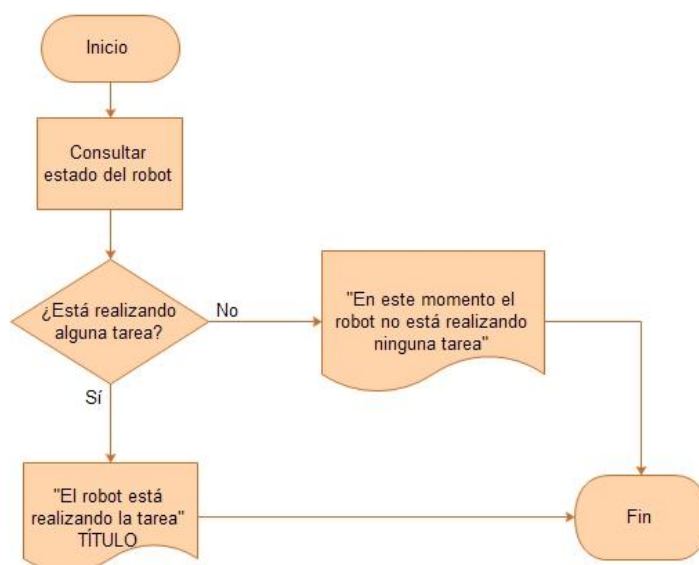
Cumpliendo con el requisito RF-10 (Reanudar tarea), en el diagrama de la derecha de la figura 4.7 se muestra el diagrama de flujo al reanudar una tarea. Actualmente, sólo se pueden reanudar las tareas de las funcionalidades historias y multimedia.

Seguidamente se describe como ejemplo, el flujo resultante cuando el usuario pulsa el botón “reanudar” para que el robot reanude una historia:

- 1- El usuario selecciona la funcionalidad historias desde la página principal o desde el menú de la aplicación.
- 2- En la aplicación se muestra la página correspondiente con la lista de historias.
- 3- El usuario pulsa el botón “reanudar” para seguir con la historia pausada.
- 4- El robot continúa contando la historia que tenía pausada.
- 5- La aplicación vuelve a la página de la lista de historias.

#### 4.2.5 Consultar estado del robot

Cumpliendo con el requisito RF-03 (Consultar estado del robot), en la figura 4.8 se muestra el diagrama de flujo al consultar el estado del robot.



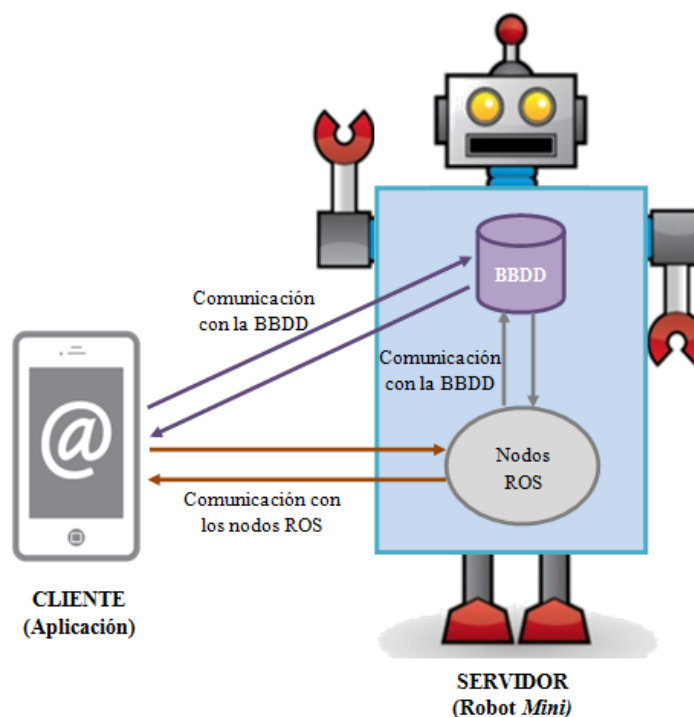
*Figura 4.8 - Diagrama de flujo: Consultar el estado del robot*

Se describe como ejemplo, el flujo resultante cuando el usuario quiere saber qué tarea está realizando el robot.

- 1- El usuario pulsa el botón “Inicio/Estado” del menú para consultar el estado del robot.
- 2- La aplicación carga la página principal.
- 3- El usuario puede ver si el robot está realizando o no una tarea.

## 4.3 Esquema del sistema

Para conseguir la funcionalidad de la aplicación representada en los diagramas de flujo del apartado 4.2, se hace uso de un sistema distribuido cliente – servidor cuyo esquema se muestra en la figura 4.9.



*Figura 4.9 - Esquema general del sistema*

En dicho sistema, el cliente es la aplicación a desarrollar y el servidor es el robot *Mini* con el que se debe conectar. Para mostrar en las páginas de cada funcionalidad de la aplicación la tabla con la lista de tareas disponibles, es necesario que la aplicación se comuniquen con la base de datos (BBDD) que tiene implementada el robot *Mini*. Por otro lado, para que éste realice, pare, pause y reanude una tarea y devuelva su estado actual, es necesario que la aplicación se comuniquen con los nodos ROS del robot *Mini*.

# Capítulo 5

## Implementación de la versión 1.0

En este capítulo se detallan las decisiones de implementación para desarrollar la primera versión de la aplicación, se describen las tecnologías, librerías y herramientas utilizadas en ella, así como cada uno de los bloques del esquema del sistema y los diagramas de secuencia para cada caso de uso definidos en el capítulo 3.

## 5.1 Aplicación web (v.1.0)

En base a los objetivos del proyecto que se desean alcanzar, a los requisitos y casos de uso del capítulo 3 y al esquema del sistema del apartado 4.3, se elige el tipo de aplicación móvil a implementar tras realizar un análisis de las características de cada tipo de aplicación disponible en la actualidad.

### 5.1.1 Tipos de aplicaciones móviles

En esta sección se realiza una comparativa de los diferentes tipos de aplicaciones móviles. Se pueden diferenciar entre tres tipos: aplicación nativa, web e híbrida.

#### 5.1.1.1 Aplicación Nativa

Una aplicación nativa es una aplicación software desarrollada bajo el lenguaje nativo y entorno de desarrollo del dispositivo de destino. Esto permite que su funcionamiento sea muy fluido y estable para el SO para el que fue creada.

Este tipo de aplicaciones permiten explotar al máximo las características del dispositivo. Aunque, debido a que el lenguaje de la aplicación debe ser el lenguaje nativo del dispositivo, una aplicación diseñada para una plataforma, por ejemplo Android, no se puede instalar en un dispositivo con otro tipo de plataforma, como por ejemplo en un iPhone. En el caso de querer utilizar la aplicación en más plataformas se debe crear una aplicación por cada plataforma en el lenguaje de programación del SO seleccionado, lo que incrementa el coste de desarrollo y de mantenimiento de este tipo de aplicaciones para múltiples SOs.

Las aplicaciones nativas proporcionan una gran experiencia de usuario porque ofrecen un rendimiento más rápido y acceso directo a los servicios del dispositivo (gestionar el calendario, los contactos, álbumes de fotos, hacer llamadas telefónicas, enviar/recibir mensajes de texto, etc.). No obstante, para conseguir este rendimiento se requiere crear recursos adicionales para el desarrollo (imágenes, segmentos de audio y diversos archivos de declaración específicos del SO) y compilar el código fuente para posteriormente distribuir la aplicación.

Este tipo de aplicaciones son distribuidas en los mercados de aplicaciones móviles como el *Google Play* para Android y el *Apple App Store* para iOS, facilitando el proceso de marketing y promoción de la aplicación.

Son instaladas en el sistema de archivos de cada dispositivo utilizando el SDK (*Software Development Kit* – Kit de Desarrollo de Software) específico de la plataforma y las herramientas propias de cada SO móvil. Por esta razón, para que el usuario disponga de la última versión debe descargarse la aplicación ya actualizada del mercado de aplicaciones y volver a instalarla en el dispositivo.

En la tabla 5.1 se indican los fabricantes, lenguajes, herramientas, formato y canales de distribución de los principales SOs móviles.



SO del dispositivo	Android	iOS	Windows Phone	Blackberry
<b>Fabricante</b>	Google	Apple	Microsoft	RIM ( <i>Research In Motion</i> – Investigación en Movimiento)
<b>Lenguaje</b>	Java (algunos C, C++)	Objective-C, C, C++	C# (C Sharp), VB.NET, etc.	Java
<b>Herramientas</b>	Android SDK	Xcode	Visual Studio, Windows Phone	BB Java Eclipse <i>Plug-in</i>
<b>Formato</b>	.apk	.app	.xap	.cod
<b>Tiendas</b>	<i>Google Play</i>	<i>Apple App Store</i>	<i>Windows Phone Marketplace</i>	<i>Blackberry App World</i>

*Tabla 5.1 - Algunas características de SOs móviles*

Una vez definidas las características de las aplicaciones nativas, el primer paso es determinar la plataforma y SO para el que se va a desarrollar. Según el análisis realizado en el apartado 2.5, el SO que debería escogerse sería Android debido a que, actualmente, es el dominante por su alta cuota de mercado. Esta elección permitirá que la aplicación pueda ser utilizada por el mayor número de usuarios. Una vez que se ha elegido el SO y disponemos de las herramientas necesarias, el siguiente paso es analizar las bibliotecas cliente que ofrece ROS (subapartado 2.4.2) y escoger aquella que se adapte a nuestras necesidades. Dado que el sistema operativo elegido para este tipo de aplicaciones es Android, la biblioteca cliente a utilizar sería *rosjava* [9].

Uno de los objetivos principales de *rosjava* es llevar ROS a Android. *Rosjava* es una implementación en Java puro de ROS. Se trata de una biblioteca cliente que permite a los programadores de Java interactuar de forma rápida con los temas, servicios y parámetros de ROS. Es de tipo asíncrono y sólo proporciona tutoriales de instalación para la distribución de *ROS Hydro*, dejando sin documentación a las distribuciones de ROS anteriores. Actualmente se encuentra en estado experimental ya que está en fase de desarrollo, de forma que las API (*Application Programming Interface* – Interfaz de Programación de Aplicaciones) y documentación proporcionadas son volátiles.

En la tabla 5.2 se recogen las ventajas e inconvenientes más relevantes de este tipo de aplicaciones.

Ventajas	Inconvenientes
1) Acceso completo al hardware del dispositivo	1) Una aplicación por plataforma y dispositivo. Código no reutilizable.
2) Explotación máxima de las características del dispositivo	2) Diferentes lenguajes de programación y herramientas de desarrollo para cada plataforma
3) Mayor rendimiento	3) Se requiere de recursos adicionales
4) Acceso directo a los servicios del dispositivo	4) Incremento de coste de desarrollo y mantenimiento de aplicaciones para múltiples SOs.
5) Distribución en los mercados de aplicaciones	5) Actualización constante de la aplicación
6) Acceso directo a la aplicación mediante un icono	6) Se requiere aprobación externa para su publicación y actualización

*Tabla 5.2 - Ventajas e inconvenientes: Aplicaciones nativas*

#### 5.1.1.2 Aplicación Web Móvil

Las aplicaciones web móvil están optimizadas para ser visualizadas en el navegador del dispositivo móvil independientemente de la plataforma y SO de éste. De forma que, al ser multiplataforma, no es necesario escoger el SO para el cual se desarrollará la aplicación como en el caso de las nativas. Por tanto, este tipo de aplicaciones no tienen la misma potencia, rapidez y recursos que una nativa y requieren utilizar el navegador web. Sin embargo, el coste inicial de desarrollo es menor que el de las aplicaciones nativas ya que no es necesario crear una aplicación para cada tipo de plataforma.

Se ejecutan dentro del propio navegador web del dispositivo a través de una URL (*Uniform Resource Locator* – Localizador Uniforme de Recursos) adaptándose el contenido al tamaño de la pantalla del mismo (*responsive*). No requieren instalación en el dispositivo por lo que se promocionan y comercializan de forma independiente sin mercado de aplicaciones.

La comunicación con el robot se hará por medio de *rosbridge* [10], debido a que permite realizar la comunicación entre un sistema ROS y un cliente remoto. Para ello, *rosbridge* proporciona una interfaz JSON (*JavaScript Object Notation* – Notación de Objetos de JavaScript) [11] que permite a cualquier cliente enviar mensajes JSON, publicar o suscribirse a temas de ROS, llamar a servicios de ROS, etc.

Los lenguajes utilizados en este tipo de aplicaciones son: HTML (*HyperText Markup Language* – Lenguaje de Marcado de Hipertexto), JavaScript y CSS (*Cascading Style Sheets* – Hojas de Estilo en Cascada), descritos más adelante, en el apartado 5.2. De modo que para interactuar con ROS desde el navegador web, se hace uso de la biblioteca JavaScript *roslibjs* [12]. Dicha librería utiliza *WebSockets* para realizar una conexión bidireccional entre el cliente web y el servidor de *rosbridge*.

En la tabla 5.3 se recogen las ventajas e inconvenientes más relevantes de este tipo de aplicaciones.

Ventajas	Inconvenientes
1) Reutilización de código fuente. Misma aplicación para múltiples plataformas	1) Acceso limitado a elementos y características del hardware del dispositivo
2) Fácil uso y mayor rango de usuarios	2) Menor tiempo de respuesta
3) Bajo coste de desarrollo	3) Distribución independiente. Mayor esfuerzo en promoción y visibilidad
4) No es necesaria la aprobación externa para publicarse	4) Requiere el uso de un navegador web
5) Reutilización de sitios <i>responsive</i> ya diseñados	
6) El usuario dispone siempre de la última versión. Menor coste de mantenimiento	

*Tabla 5.3 - Ventajas e inconvenientes: Aplicaciones web*

### 5.1.1.3 Aplicación Híbrida

Como combinación de las aplicaciones nativas y web explicadas anteriormente, surgen las aplicaciones híbridas incorporando las ventajas de ambas.

Este tipo de aplicaciones se desarrollan con los lenguajes de las aplicaciones web (HTML, CSS y JavaScript) permitiendo así su uso en cualquier plataforma y dispositivo. Además es posible acceder a gran parte de las características del hardware de éste y puede ser distribuida en el mercado de aplicaciones propio del dispositivo.

La comunicación con ROS también se realiza mediante *roslibjs* de *rosbridge* de igual manera que para las aplicaciones web.

Para facilitar el desarrollo de estas aplicaciones híbridas hay disponible diversos *frameworks* como PhoneGap/Cordova (ver apartado 5.1.3.2).

En la tabla 5.4 se recogen las ventajas e inconvenientes más relevantes de este tipo de aplicaciones.

Ventajas	Inconvenientes
1) Reutilización de código fuente.	1) Menor tiempo de respuesta y acceso a los servicios del dispositivo
2) Construida con lenguajes de desarrollo web: HTML, JavaScript y CSS	
3) Acceso directo a la aplicación mediante un icono	
4) Acceso a parte del hardware del dispositivo	

*Tabla 5.4 - Ventajas e inconvenientes: Aplicaciones híbridas*

### 5.1.2 Comparativa de los tipos de aplicaciones

Para poder realizar el diseño y por tanto la implementación de la aplicación, es necesario decidir en primer lugar el tipo de aplicación que se va a desarrollar teniendo en cuenta los requisitos y casos de uso definidos, ya que será la base para tomar el resto de decisiones.

#### Aplicación nativa

- Si no es prioritario que la aplicación sea multiplataforma y se utilice un único SO móvil.
- Si es necesario utilizar alguna de las características del hardware del dispositivo.
- Si se requiere de una gran experiencia de usuario, es decir, que haga uso de muchas funciones y capacidad de respuesta en tiempo real. Por ejemplo, para un juego.
- Si se prevé que será rentable ya que el coste de desarrollo y mantenimiento es mayor que para otro tipo de aplicaciones.

#### Aplicación Web

- Si el requisito es que la aplicación sea multiplataforma, es decir, que se pueda utilizar en cualquier dispositivo y SO.
- Si se necesita crear la aplicación de forma rápida y poco costosa, ya que los costos y tiempo de lanzamiento del desarrollo es menor que el de una nativa.

#### Aplicación Híbrida

- Si se requiere una aplicación multiplataforma que tenga acceso a alguna de las características y funcionalidades del dispositivo
- Si se requiere que el proceso de desarrollo y de mantenimiento sea rápido y de bajo coste.

En la tabla 5.5 se muestran algunas de las características para comparar las aplicaciones móviles nativa, web e híbrida.

Característica	Aplicación Nativa	Aplicación Web	Aplicación Híbrida
<b>Lenguaje de desarrollo</b>	Nativo	Web	Nativo y Web o sólo nativo
<b>Experiencia de instalación</b>	Alta Desde mercado de aplicaciones	Mediana Desde navegador móvil	Alta Desde mercado de aplicaciones
<b>Portabilidad</b>	Bajo	Alto	Alto
<b>Acceso a características del dispositivo</b>	Alto	Bajo	Mediano
<b>Gráficos avanzados</b>	Alto	Mediano	Mediano
<b>Actualizaciones</b>	Bajo Desde mercado de aplicaciones	Alto	Mediano Frecuentemente desde mercado de aplicaciones

*Tabla 5.5 - Comparativa de los tipos de aplicaciones*

Valorando las ventajas e inconvenientes de cada tipo de aplicación y teniendo en cuenta la comunicación entre la aplicación y ROS, se ha decidido desarrollar la **aplicación web** por las siguientes razones:

- Debido a que no se requiere gran experiencia de usuario, ni emplear el hardware del dispositivo para implementar las funcionalidades indicadas (música, fotos, tv/videos, noticias, historias y multimedia), no es imprescindible que la aplicación sea nativa.
- *Rosjava* aún se encuentra en fase experimental, mientras que la comunicación entre el cliente web y ROS mediante *rosbridge* con *roslibjs* es sencilla y estable.
- Al desarrollar una aplicación para un SO específico, se limita el uso de la aplicación a dicha plataforma. Sin embargo una aplicación web puede ser utilizada por cualquier dispositivo con conexión a Internet y un navegador.
- Dependiendo de cómo se desarrolle la aplicación web cabe la posibilidad de hacer una aplicación híbrida para el SO que se escoja.
- La aplicación web no requiere ningún software adicional, sin embargo tanto para el desarrollo de la aplicación nativa como híbrida para Android, es necesario obtener el JDK (*Java Development Kit* - Kit de Desarrollo de Java), descargar el software Eclipse con el *plugin* ADT (*Eclipse with Android Developer Tools* – Eclipse con Herramientas de Desarrollo Android) y el SDK de Android para disponer de las librerías API y de las herramientas de desarrollo para crear, probar y depurar la aplicación.

### 5.1.3 Frameworks

En la valoración realizada en el subapartado anterior el tipo de aplicación escogido es web, de modo que es muy recomendable utilizar un *framework* para reducir el tiempo, facilitar y simplificar el desarrollo de la aplicación.

Un *framework* es un conjunto estandarizado de conceptos y criterios para llevar a cabo un problema común. Debido a que la mayoría de aplicaciones web comparten una estructura muy similar, el objetivo principal de los *frameworks* es ofrecer dicha estructura común para reducir el tiempo en el proceso de desarrollo reutilizando código ya existente.

Para el caso de aplicaciones web e híbridas hay disponible diversidad de *frameworks* que permiten al desarrollador crear una aplicación atractiva para el usuario de manera sencilla y rápida.

A continuación se explicarán algunas de las características de varios *frameworks* seleccionados para cada tipo de aplicación, cuyo objetivo principal es similar. Sin embargo difieren en ciertas características. Las razones de dicha elección ha sido escoger algunos de los *frameworks* para *smartphones* disponibles en la actualidad que permitan cumplir con los requisitos del proyecto.

### 5.1.3.1 Frameworks para aplicaciones Web

Para el desarrollo de aplicaciones web se detallan las características de algunos *frameworks* CSS debido a que hacen posible el desarrollo web más rápido y fácil, proporcionando una serie de clases CSS con los elementos necesarios para la aplicación.

#### a) Twitter Bootstrap 3



Bootstrap [13] es un *framework* bastante popular, intuitivo y potente para el desarrollo web sencillo y rápido. Está diseñado para ser utilizado por personas de todos los niveles, en dispositivos de diversos tamaños y de diferentes SOs y en proyectos de cualquier magnitud.

Inicialmente fue desarrollado para navegadores de escritorio al que se le ha añadido soporte para móvil permitiendo la adaptación a cualquier tamaño de pantalla (*responsive*). Es un *framework* de código abierto construido por Twitter y se gestiona a través de GitHub [14].

Bootstrap ofrece el código fuente CSS para poder modificarlo y el CSS precompilado para facilitar los procesos de descarga y utilización.

Dispone de una amplia documentación para los elementos comunes de HTML, componentes CSS y HTML personalizados.

A través de la página web de *Bootswatch* [15], se pueden obtener diferentes plantillas de ejemplo de modo gratuito para personalizar los elementos de la aplicación. Y mediante la extensión de Bootstrap, *Font-Awesome* [16], es posible personalizar diversidad de iconos vectoriales escalables.

Este *framework* está bajo la licencia MIT (*Massachusetts Institute of Technology* – Instituto Tecnológico de Massachusetts) que requiere incluir la licencia y el aviso de copyright en la aplicación.

#### b) Foundation 5



Foundation [17] es un *framework* desarrollado por la empresa de diseño de productos, ZURB. Es un *framework* avanzado y *responsive*, que permite la creación de sitios y aplicaciones de una manera rápida y sencilla compatible con todos los dispositivos y la mayoría de navegadores.

El diseño de un sitio web o aplicación se realiza en primer lugar para el dispositivo pequeño (*Mobile First*), es decir, para el *smartphone* y posteriormente se expande a tamaños de pantallas superiores. Es un *framework* apoyado profesionalmente por una organización (ZURB) y ha sido probado durante años.

ZURB proporciona un conjunto de cursos, videos, bloques de construcción (trozos de código reutilizables) y una documentación bastante extensa con ejemplos de uso y de código.

La última versión de Foundation (versión 5) da mayor prioridad al aumento de la velocidad para lograr obtener una mejor experiencia de usuario, y crear sitios web más rápidos y optimizados. Por esta razón, esta versión incorpora la tecnología *Interchange* que hace posible cargar de forma selectiva secciones enteras en función del tipo del dispositivo del cliente.

Al igual que Bootstrap, Foundation está bajo la licencia MIT por lo que se requiere que se incluya la licencia y el aviso de copyright en la aplicación.

### c) Skeleton



Es un *framework* ligero y fácil de personalizar [18]. Está compuesto por una colección de archivos CSS para realizar un desarrollo rápido de sitios web para cualquier dispositivo. Está basado en tres principios fundamentalmente:

- Cuadrícula *responsive*: está basado en una rejilla con un ancho máximo de 960 píxeles y se reduce proporcionalmente al tamaño de las pantallas de dispositivos más pequeños.
- De inicio rápido: está orientado para el desarrollo rápido de sitios web.
- No es un *framework* de interfaz de usuario. El objetivo es proporcionar como base los estilos básicos y posteriormente adoptar cualquier diseño.

También está bajo la licencia MIT.

### 5.1.3.2 Frameworks para aplicaciones Híbridas

#### a) jQuery Mobile



Aunque originalmente jQuery fue diseñado para dispositivos de escritorio, el *framework* jQuery Mobile [19] incorpora una interfaz móvil muy sencilla de usar. Se basa en “escribir menos, hacer más”, es decir, con el mismo código base HTML, permite crear aplicaciones únicas para cada tipo de dispositivo móvil y SO.

Ofrece una herramienta denominada *Constructor* para la construcción de un paquete personalizado que contiene los elementos necesarios. El objetivo de este constructor es personalizar la velocidad generando un archivo JavaScript, y las hojas de estilo CSS necesarias.

Es compatible con otras aplicaciones y *frameworks* para desarrollar aplicaciones jQuery Mobile como PhoneGap, Wijmo, Codiqa, etc.



Proporciona una amplia documentación y códigos de ejemplo para ayudar a realizar el desarrollo de la aplicación.

Este *framework* facilita al desarrollador crear temas personalizados gracias a *ThemeRoller* [20] ya que permite seleccionar los colores de cada elemento HTML y posteriormente descargar el tema personalizado.

También está bajo la licencia MIT.

#### b) PhoneGap de Adobe y Apache Cordova



PhoneGap

Phonegap [21] es un *framework* libre y de código abierto que permite la creación de aplicaciones para dispositivos móviles utilizando tecnologías web: HTML, CSS y JavaScript. Con un solo código base hace posible crear aplicaciones multiplataforma reduciendo al mínimo el tiempo de producción.

Phonegap es una distribución de Apache Cordoba. Ambos son de código abierto y se utilizan del mismo modo. Están orientados al desarrollo de aplicaciones híbridas ya que es posible acceder al hardware del dispositivo utilizando los estándares web: acelerómetro, cámara, brújula, contactos, expediente, geolocalización, medios de comunicación, red, notificaciones y almacenamiento.

Lo que diferencia a ambos *frameworks* es que Adobe puede extender el producto e incluir herramientas o características propias por las que pueda cobrar. Cordova es el producto base por lo que las actualizaciones de éste suelen ser más frecuentes. Mientras que Phonegap ofrece el servicio *PhoneGap Build* que permite la compilación en la nube, sin necesidad de disponer del SDK nativo ni de las herramientas específicas de cada plataforma. PhoneGap se encarga de compilar la aplicación desarrollada en la nube utilizando el SDK más actualizado para la plataforma destino escogida. Para proyectos de código abierto el servicio es gratuito. Dicha versión gratuita tiene como limitación un tamaño máximo de aplicación de 50 MB y *plugins* no actualizados.

Disponen de una guía de plataformas para poder configurar el entorno de desarrollo de cada una (Amazon Fire OS, Android, BlackBerry 10, iOS, Windows Phone 8, Windows y Tizen), es decir, para descargar el SDK, configurar los emuladores de dispositivos, la conexión de dispositivos para la prueba directa y para gestionar la firma de los requisitos clave. También disponen de guías adicionales con información sobre las instrucciones para añadir *plugins*, las opciones de configuración y la actualización de cada plataforma y un conjunto de herramientas de línea de comandos específicos de cada una.

Ambos *frameworks* están bajo la licencia Apache versión 2.0 de software libre creada por la ASF (*Apache Software Foundation*). Esta licencia permite al usuario utilizar el software, distribuirlo y modificarlo. Además, no exige que las versiones modificadas de dicho software se distribuyan usando la misma licencia ni como software de código abierto. Como único requisito se pide que los receptores sean informados de que se ha utilizado código con la licencia Apache.



## c) Sencha Touch



Sencha [22] es un *framework* comercial para el desarrollo de aplicaciones móviles en HTML de alto rendimiento, que consigue reducir las diferencias entre una aplicación web y una nativa. Permite crear aplicaciones multiplataforma de gran alcance, aprovechando la aceleración de hardware, independientemente del tipo de navegador o dispositivo móvil.

Es compatible con la API de Apache Cordova para tener acceso al acelerómetro, cámara, brújula, conexión, contactos, dispositivos, eventos, geolocalización, almacenamiento, notificación, etc. del dispositivo. También es compatible con *Adobe PhoneGap Build*.

Proporciona un conjunto de documentación en vídeo y otros recursos para que los nuevos desarrolladores puedan crear una aplicación de forma sencilla y rápida. Sin embargo para el desarrollo de aplicaciones con fines comerciales, es necesario comprar una licencia.

Este *framework* está bajo la licencia GPLv3 (*GNU General Public Licence v3* – Licencia Pública General de GNU versión 3). De todos los términos de esta licencia, cabe destacar que el desarrollador debe proporcionar el código fuente de la aplicación a sus usuarios para que éstos puedan modificarla en función de sus necesidades.

#### 5.1.4 Comparativa de los tipos de *frameworks*

Una vez que se han detallado algunas de las características de cada *framework*, en la tabla 5.6 y tabla 5.7 se recogen las más relevantes para realizar una comparación entre los *frameworks* seleccionados.

	App Web	App Híbrida	Optimizado para dispositivo	Alto rendimiento	Documentación
Twitter Bootstrap 3	✓	X	De escritorio con soporte para móvil	X	Amplia
Foundation 5	✓	X	<i>Mobile First</i>	X	Extensa
Skeleton	✓	X	Móvil y de escritorio	X	Reducida
jQuery Mobile	✓	✓	De escritorio con incorporación de interfaz para móvil	✓	Extensa
Sencha Touch	✓	✓	Móvil y de escritorio	✓	Extensa con herramientas y soporte
PhoneGap/ Cordova	✓	✓	Móvil	✓	Extensa

Tabla 5.6 - Comparativa *frameworks* a)

	Fácil	Conocimiento Lenguajes	Personalización	Herramientas auxiliares	Licencia
<b>Twitter Bootstrap 3</b>	✓	HTML	<i>Bootswatch, Font Awesome, etc.</i>	-	MIT
<b>Foundation 5</b>	✓	HTML, CSS y JavaScript	-	<i>Interchange</i>	MIT
<b>Skeleton</b>	✓	HTML y CSS	-	-	MIT
<b>jQuery Mobile</b>	✓	HTML	<i>ThemeRoller</i>	<i>Constructor</i>	MIT
<b>Sencha Touch</b>	X	HTML y JavaScript	-	-	GPL v3
<b>PhoneGap</b>	✓	HTML, CSS y JavaScript	-	<i>Adobe PhoneGap Build</i>	Apache v.2.0
<b>Cordova</b>	✓	HTML, CSS y JavaScript	-	-	Apache v.2.0

*Tabla 5.7 - Comparativa frameworks b)*

Valorando las diferentes características de los *frameworks* estudiados, se ha decidido utilizar el *framework* **Twitter Bootstrap 3** por ser uno de los más populares y sencillos de utilizar por personas de cualquier nivel y por proporcionar una documentación amplia de cada uno de los elementos HTML.

## 5.2 Tecnologías utilizadas

A día de hoy, hay una gran variedad de tecnologías, librerías, herramientas y estilos para desarrollar una aplicación web. En este apartado se van a diferenciar las tecnologías empleadas en el lado del cliente de las utilizadas en el lado del servidor.

En el anexo C se describen las tecnologías auxiliares utilizadas para facilitar el proceso de desarrollo de la aplicación.

### 5.2.1 Tecnologías de cliente

Las tecnologías de cliente están basadas en HTML, JavaScript y CSS. Dentro de estas tecnologías se encuentran aquellas que permiten la comunicación con el servidor y la creación de interfaces de usuario atractivas.

#### 5.2.1.1 HTML

El lenguaje HTML se emplea para crear páginas web. Se trata de un estándar definido por el W3C (acrónimo de *World Wide Web Consortium*) [23] que permite describir la estructura básica y el contenido (texto, imágenes, videos, etc.) de una página web.

El lenguaje HTML es muy sencillo, por lo que no es necesario utilizar un editor particular, sólo se necesita un editor de texto ASCII (acrónimo de *American Standard Code for Information Interchange*). Está basado en etiquetas (del inglés: *tags*), las cuales se delimitan entre los signos menor que y mayor que (< , >) que sirven para determinar la forma en la que va a aparecer el contenido en el navegador. El código HTML utiliza la siguiente sintaxis:

- 1- Etiqueta inicial que indica el comienzo de un elemento
- 2- Atributos y sus valores asociados
- 3- Contenido
- 4- Etiqueta final (puede ser opcional para algunos elementos)

La versión actual es HTML5 que aunque continúa en desarrollo, la mayoría de navegadores incorporan gran parte de su especificación. Al tratarse de un estándar web, una página web escrita en una versión determinada puede ser interpretada por cualquier navegador. No obstante, las versiones HTML han ido incorporando y eliminando características para mejorar la eficiencia y facilitar el desarrollo de páginas web compatibles con diferentes navegadores y plataformas. Para una correcta interpretación de las páginas web es necesario que el navegador incorpore dichos cambios ya que de lo contrario se pueden producir problemas de visualización.

### 5.2.1.2 CSS

CSS [24] es un lenguaje de estilo para especificar el diseño de páginas web, dando formato al contenido previamente estructurado de los documentos HTML. Permite definir las fuentes, contornos, colores, márgenes, dimensiones, fondo, bordes, tablas, etc. de una página web.

Al principio se crearon nuevas etiquetas HTML para dar formato al contenido, sin embargo todos los navegadores no incorporaban dichos cambios. Para solucionar este problema se creó el lenguaje CSS que además de separar la presentación del contenido del documento, está soportado por todos los navegadores actualmente.

La principal ventaja de la utilización de ficheros CSS es controlar la presentación de muchos documentos HTML desde una única hoja de estilo, de manera que el mantenimiento del sitio web sea más fácil y rápido.

Hay tres métodos para insertar una hoja de estilos en un documento HTML: entre líneas, dentro de la sección *head* y utilizando hojas de estilo externas. De los tres, el más recomendable es el externo ya que creando un único fichero con extensión .css se puede incluir en varios documentos HTML de manera que los cambios que se realicen en este fichero afectarán a todos los documentos HTML que lo incluya.

A continuación se muestra la manera de incluir una hoja de estilos externa en un documento HTML:

```
<head>
...
<link rel="stylesheet" type="text/css" href="/archivo.css">
...
</head>
```

### 5.2.1.3 JavaScript

Se trata de un lenguaje interpretado, es decir, el navegador web (cliente) es el encargado de analizar y procesar las instrucciones en el momento que deben ser ejecutadas. También se conoce como lenguaje *script*, es decir, es el código de programación que se inserta dentro de un documento. Por esta razón, no son necesarios procesos intermedios, los programas escritos en JavaScript se pueden probar directamente en el navegador.

Se utiliza principalmente en el lado del cliente, donde los programas son ejecutados por el navegador web. No obstante, también existe JavaScript del lado del servidor para uso en aplicaciones externas a la web.

Este lenguaje se utiliza para crear páginas web dinámicas, es decir, páginas que incorporan efectos (por ejemplo, texto que aparece y desaparece), animaciones, pop-ups con mensajes para avisar al usuario, etc.

Debido a su nombre podría suponerse que está relacionado con el lenguaje de programación Java, sin embargo no tienen nada que ver. JavaScript, al contrario que Java, no crea programas independientes sino que dependen del código HTML de la página web. El código JavaScript se incrusta dentro del código HTML, de manera que un programa JavaScript debe ir encerrado con la etiqueta “<script>” y la propiedad *type* debe ir inicializada con la cadena “text/javascript” de la siguiente forma:

```
<script type="text/javascript">
... código JavaScript ...
</script>
```

Al igual que los ficheros .css, en JavaScript también es posible crear un fichero con extensión .js que contenga las funciones JavaScript necesarias para la aplicación e incluirla en el *head* de cada documento HTML, de manera que sea más rápido y sencillo de mantener, ya que es común a todas las páginas web de la aplicación.

```
<head>
...
<script type="text/javascript" src="/archivo.js"></script>
...
</head>
```

### 5.2.1.4 jQuery

jQuery [25] es una biblioteca de JavaScript rápida y ligera, de software libre y de código abierto, que está bajo la Licencia MIT y la Licencia Pública General de GNU v2. Su principal objetivo es facilitar el desarrollo de aplicaciones web en el lado del cliente. jQuery no es un lenguaje sino un conjunto de funciones y métodos JavaScript incluidos en un único fichero JavaScript.

Incorpora las funcionalidades de DOM (*Document Object Model*), eventos, efectos y AJAX (*Asynchronous JavaScript and XML (eXtensible Markup Language)* – JavaScript Asíncrono y XML), de modo que las ventajas que ofrece jQuery son las siguientes:

- Se requieren menos líneas de código → desarrollo más rápido y sencillo
- Se encarga de que el código JavaScript de la aplicación sea compatible con cualquier navegador web.
- Proporciona un mecanismo para manejar eventos.
- Incorpora una serie de funciones que nos permite animar el contenido de la página de manera sencilla.
- Integra funcionalidades para interaccionar con la técnica AJAX (explicada más adelante, en el apartado 7.1.2.1)

### 5.2.1.5 Twitter Bootstrap

Como se ha detallado en el apartado 5.1.3.1 (a), Bootstrap es un *framework* CSS para desarrollar aplicaciones web compatibles con todas las plataformas.

Para utilizar este *framework* se han descargado los ficheros CSS, JavaScript y las fuentes compiladas de Bootstrap de [13] y se ha incluido lo siguiente en las páginas web:

```
...  
<head>  
...  
    <link rel="stylesheet" href="../../bootstrap/css/bootstrap.min.css">  
...  
</head>  
<body>  
...  
    <script src="../../bootstrap/js/bootstrap.min.js"></script>  
...  
</body>  
...
```

En la figura 5.1 se muestran unos ejemplos de las plantillas gratuitas que ofrece *Bootswatch* para este *framework*.

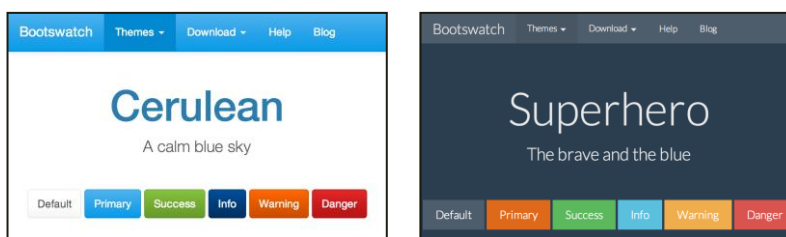


Figura 5.1 - Plantillas gratuitas de Bootswatch



### 5.2.1.6 Font-Awesome

Se trata de una fuente de iconos web que proporciona una gran colección de iconos vectoriales escalables que pueden ser personalizados en tamaño, color, sombra, y en todo lo que es posible realizar con CSS. Originalmente fue diseñado para Bootstrap aunque también trabaja bien con otros *frameworks*. Se caracteriza por ser simple, por no

requerir JavaScript y ser de código abierto bajo la licencia GPL. Permite su uso en proyectos comerciales, de código abierto, entre otros.

Para incluir los iconos en la página web se ha descargado el proyecto de *Font-Awesome* de [16] y se ha incluido lo siguiente en las páginas web:

```
...
<head>
    ...
    <link rel="stylesheet" href="../../font-awesome/css/font-awesome.min.css">
    ...
</head>
<body>
    ...
    <i class="fa fa-music"></i>
    ...
</body>
...
```

Donde “fa-music” es el nombre del icono . Para utilizar otro hay que sustituirlo por el nombre del icono deseado, como por ejemplo “fa-picture-o” para mostrar el icono .

## 5.2.2 Tecnologías de servidor

Las tecnologías de servidor permiten implementar el comportamiento de la aplicación web en el servidor, es decir, la lógica de negocio, la comunicación con bases de datos, etc.

### 5.2.2.1 Apache

El equipo del robot proporciona el servidor web de software libre más popular, Apache, que utiliza una arquitectura cliente-servidor, es decir, el cliente hace una solicitud o petición al servidor y éste la atiende.

El *Apache HTTP Server (httpd)* [26] es un proyecto de ASF cuyo objetivo es desarrollar y mantener un servidor HTTP de código abierto, seguro, eficiente y extensible. Se integra fácilmente con otras aplicaciones de código abierto, tales como PHP (*PHP Hypertext Preprocessor*) y MySQL. Actualmente se puede hacer uso del paquete de código abierto que contiene MySQL, PHP y Perl denominado XAMPP [27] y acrónimo de X (para cualquier sistema operativo), Apache, MySQL, PHP, Perl.

### 5.2.2.2 PHP

PHP [28] es un acrónimo recursivo que significa *PHP Hypertext Preprocessor*. Es un lenguaje de programación de propósito general que se ejecuta en el lado del servidor cuyo objetivo es desarrollar páginas web de contenido dinámico. El código PHP es fácil de aprender y se puede introducir en las páginas HTML entre “<?php” y “>”. Cuando el servidor interpreta el código PHP, genera una salida HTML que mostrará el resultado en los navegadores que realizaron la petición.

Al ser un lenguaje que se ejecuta en el servidor es independiente del navegador web por lo que no se requiere que éste soporte PHP, pero sí es necesario que sea soportado por el servidor en el que se encuentran las páginas con contenido PHP. Además permite acceder a los recursos que tenga el servidor, como por ejemplo una base de datos.

### 5.2.2.3 C++

La biblioteca cliente de ROS que se utiliza es *roscpp* cuyo lenguaje de programación es C++ [29], de modo que se requiere este lenguaje para desarrollar los servicios y temas de ROS necesarios para la aplicación. Es considerado un lenguaje de nivel medio ya que comprende características de lenguajes de alto y bajo nivel. Se trata de una extensión del lenguaje de programación C con programación orientada a objetos, incluyendo clases, herencia básica, procesos en línea, argumentos de funciones, comprobación de tipos y todas las características del lenguaje C.

## 5.2.3 Base de datos

Las bases de datos son una parte esencial del desarrollo web. Se utilizan para almacenar y consultar la información relevante para la aplicación.

### 5.2.3.1 MySQL

MySQL [30] es un sistema de base de datos que se ejecuta en el servidor y utiliza el estándar SQL (*Structure Query Language* – Lenguaje de Consulta Estructurado). Se caracteriza por ser la mayor base de datos de código abierto bajo la licencia GPL, muy rápida, fiable y fácil de usar. Es de tipo relacional y permite crear bases de datos y tablas, modificarlas, eliminarlas, insertar datos, modificarlos, eliminarlos, ordenarlos, hacer consultas, y más operaciones.

Para conectarse al servidor es necesario un nombre de usuario y una contraseña. En caso de que el servidor se encuentre en una máquina diferente, es necesario también el nombre o dirección IP de éste.

Desde la consola de Linux podemos acceder a la base de datos del servidor ejecutando lo siguiente:

```
shell> mysql -h NombreServidor -u NombreUsuario -p
Enter password: *****
```

### 5.2.3.2 MySQL++

Para el acceso a la base de datos desde los servicios de ROS escritos en el lenguaje de programación C++ se hace uso de MySQL++ [31].

Las principales clases de cara al usuario de MySQL++ son: *mysqlpp::Connection*, *mysqlpp::Query*, *mysqlpp::Row*, *mysqlpp::StoreQueryResult* y *mysqlpp::UseQueryResult*. Además dispone del mecanismo SSQS (*Specialized SQL Structures* – Estructuras especializadas SQL) que permite crear estructuras de C++ paralelas a la definición de tablas en el esquema de base de datos.

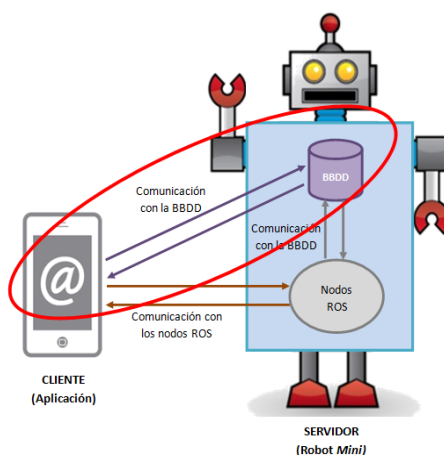
Tanto la librería MySQL++ como su manual de referencia se publican bajo LPGL (*GNU Lesser General Public License* - Licencia Publica General de GNU).

## 5.3 Comunicación cliente – servidor

Según el esquema del sistema de la figura 4.9, la aplicación se debe comunicar tanto con la base de datos MySQL como con los nodos ROS del robot *Mini*. Por tanto, este apartado se divide en dos subapartados para explicar cada tipo de comunicación y detallar cada elemento que lo compone.

### 5.3.1 Aplicación – Base de datos

En este subapartado se detallan los elementos que intervienen en la comunicación entre la aplicación y la base de datos (sección remarcada de la figura 5.2).



*Figura 5.2 - Esquema general: Aplicación – Base de datos*

#### 5.3.1.1 Arquitectura de la aplicación web

Las aplicaciones web se diferencian unas de otras en el contenido y en el aspecto gráfico. La arquitectura básica de una aplicación web sigue la estructura de las arquitecturas cliente-servidor.

La arquitectura cliente-servidor es una arquitectura distribuida en la cual los clientes solicitan servicios y los servidores los proporcionan. Está compuesta por los siguientes elementos como se muestra en el esquema de la figura 5.3:



*Figura 5.3 - Esquema básico de una aplicación web*



- Cliente: en una aplicación web, el cliente es el navegador web y tiene dos funciones:
  - a) Hace de interfaz con el usuario permitiendo atender sus peticiones, mostrando los resultados de las consultas y proporcionando un conjunto de herramientas que facilitan al usuario su comunicación con el servidor.
  - b) Transmite las peticiones de los usuarios al servidor web.
- Servidor web: recibe las peticiones de los clientes (navegadores) y las responde enviando el recurso solicitado o en caso de que dicho recurso no exista, notifica un error.
- Protocolo HTTP (*Hypertext Transfer Protocol* – Protocolo de Transferencia de Hipertexto)[32]: es un protocolo de nivel de aplicación del modelo OSI (*Open System Interconnection*) basado en TCP/IP (*Transmission Control Protocol/Internet Protocol*), aplicado para la comunicación cliente-servidor, es decir, permite al navegador realizar peticiones al servidor web y que éste las responda.
- HTML: lenguaje para crear los contenidos de la web, basado en SGML (*Standard Generalized Markup Language*).

No todas las aplicaciones web tienen la misma estructura, depende de lo estática o dinámica que sea la web. Se pueden diferenciar cuatro tipos de estructuras: cliente estático - servidor estático, cliente estático - servidor dinámico, cliente dinámico - servidor estático y cliente dinámico - servidor dinámico.

El cliente es dinámico cuando contiene código JavaScript que se ejecuta en el navegador. Mientras que el servidor es dinámico cuando se requiere ejecutar código en el servidor para generar el recurso solicitado.

Para la aplicación web que se desea implementar se debe utilizar la estructura cliente dinámico – servidor dinámico:

- Cliente dinámico: el cliente hace uso de la librería *roslibjs* para establecer conexión con *rosbridge* mediante *WebSockets*.
- Servidor dinámico: se necesita consultar la base de datos proporcionada para mostrar las tareas disponibles del robot.

Según el esquema básico de la aplicación web, las especificaciones proporcionadas y los requisitos que se deben cumplir, la arquitectura de la aplicación web a desarrollar sigue el esquema de la figura 5.4. En esta arquitectura el navegador web es el navegador del dispositivo móvil del usuario y en el equipo del robot se encuentra el servidor web junto con la base de datos proporcionada.

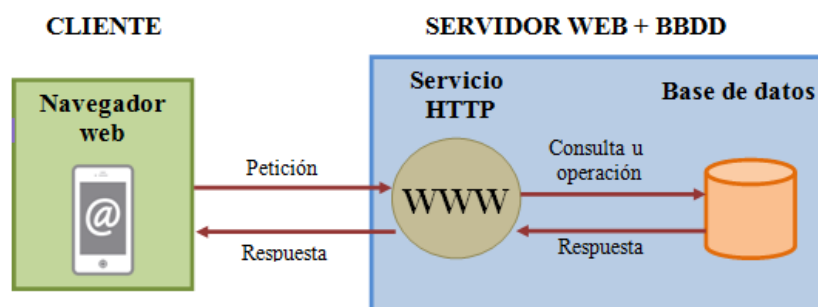


Figura 5.4 - Arquitectura de la aplicación web

Siguiendo la arquitectura cliente – servidor, en la figura 5.5 se muestran los elementos para llevar a cabo la comunicación entre el cliente (aplicación) y el servidor web (robot *Mini*).

El usuario a través de la aplicación realiza una petición al servidor para que le envíe una página web. El servidor busca la página solicitada en el repositorio de páginas y ejecuta el intérprete PHP que procesará el código de dicha página generando el contenido dinámico (consultando la base de datos) que será enviado por el intérprete PHP al servidor web, el cual se lo enviará al cliente (aplicación).

Para comprender mejor la figura 5.5 se explica el caso de uso CU-05 (Seleccionar funcionalidad) para ver las canciones disponibles que puede reproducir el robot:

- El usuario selecciona la funcionalidad música en la aplicación para visualizar las canciones disponibles.
- La aplicación solicita al servidor web la página .php correspondiente que se encuentra en el repositorio de páginas del robot. Dicha página consulta las canciones disponibles de la base de datos que tiene implementada éste.
- El servidor web devuelve la consulta de la base de datos a la aplicación en código HTML.
- La aplicación muestra al usuario una tabla con la lista de canciones disponibles que puede reproducir el robot.

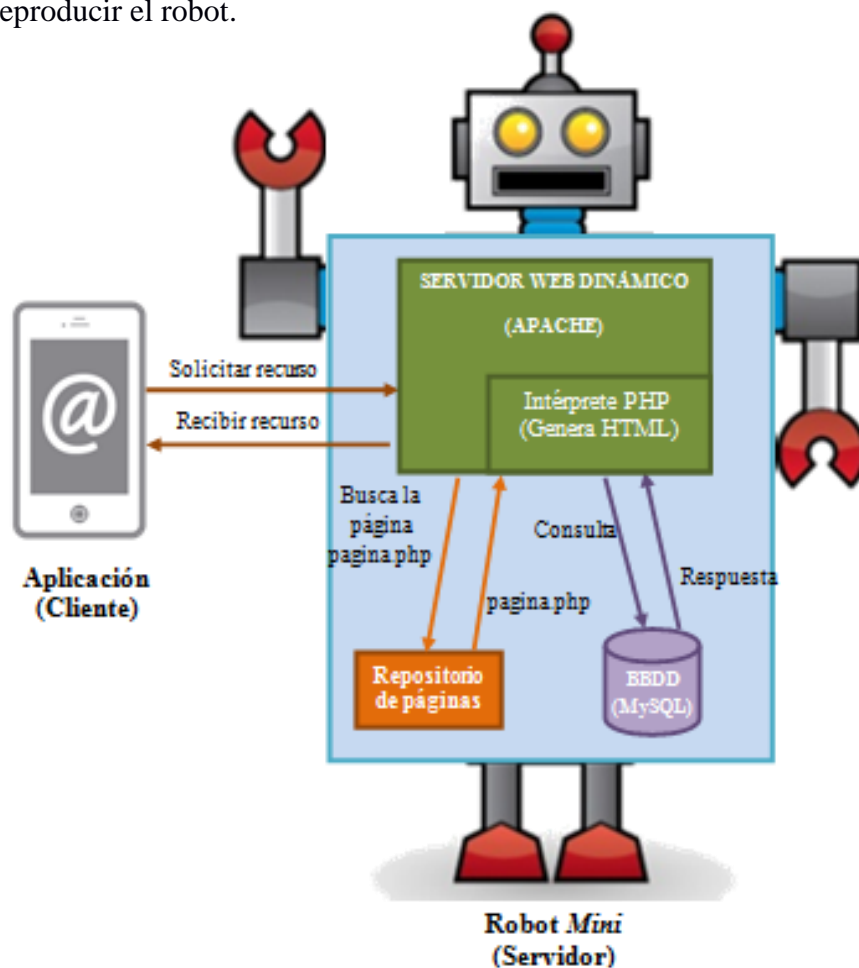
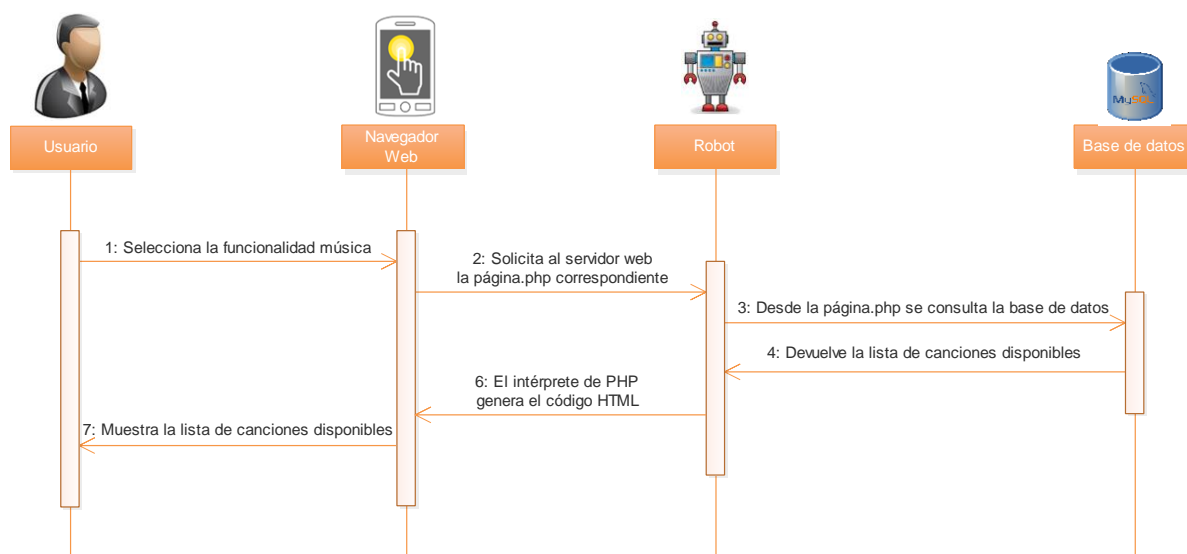


Figura 5.5 - Esquema de comunicación: Aplicación – Base de datos

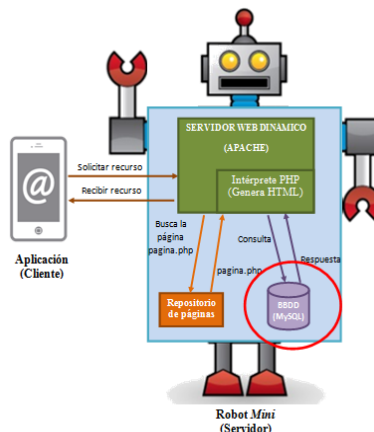
En la figura 5.6 se muestra el diagrama de secuencia del ejemplo descrito:



*Figura 5.6 - Diagrama de secuencia: Aplicación – Base de datos*

### 5.3.1.2 Base de datos

En este subapartado se detalla la base de datos MySQL del robot *Mini* (figura 5.7) que contiene la información necesaria para que el robot pueda realizar las tareas de las diversas funcionalidades de entretenimiento que va a desempeñar.



*Figura 5.7 - Esquema de comunicación: Base de datos*

La base de datos proporcionada *robalz* está formada por un conjunto de tablas y vistas MySQL. En la figura 5.8 se pueden observar las tablas disponibles y la relación entre ellas.

Según el requisito RF-05 (Seleccionar funcionalidad), la aplicación debe disponer de las funcionalidades: música, fotos, tv/videos, noticias, historias y multimedia. El contenido necesario para dichas funcionalidades está almacenado en las tablas “multimedia\_content” y “stories” y en las vistas “multimedia\_tags” y “tags”.

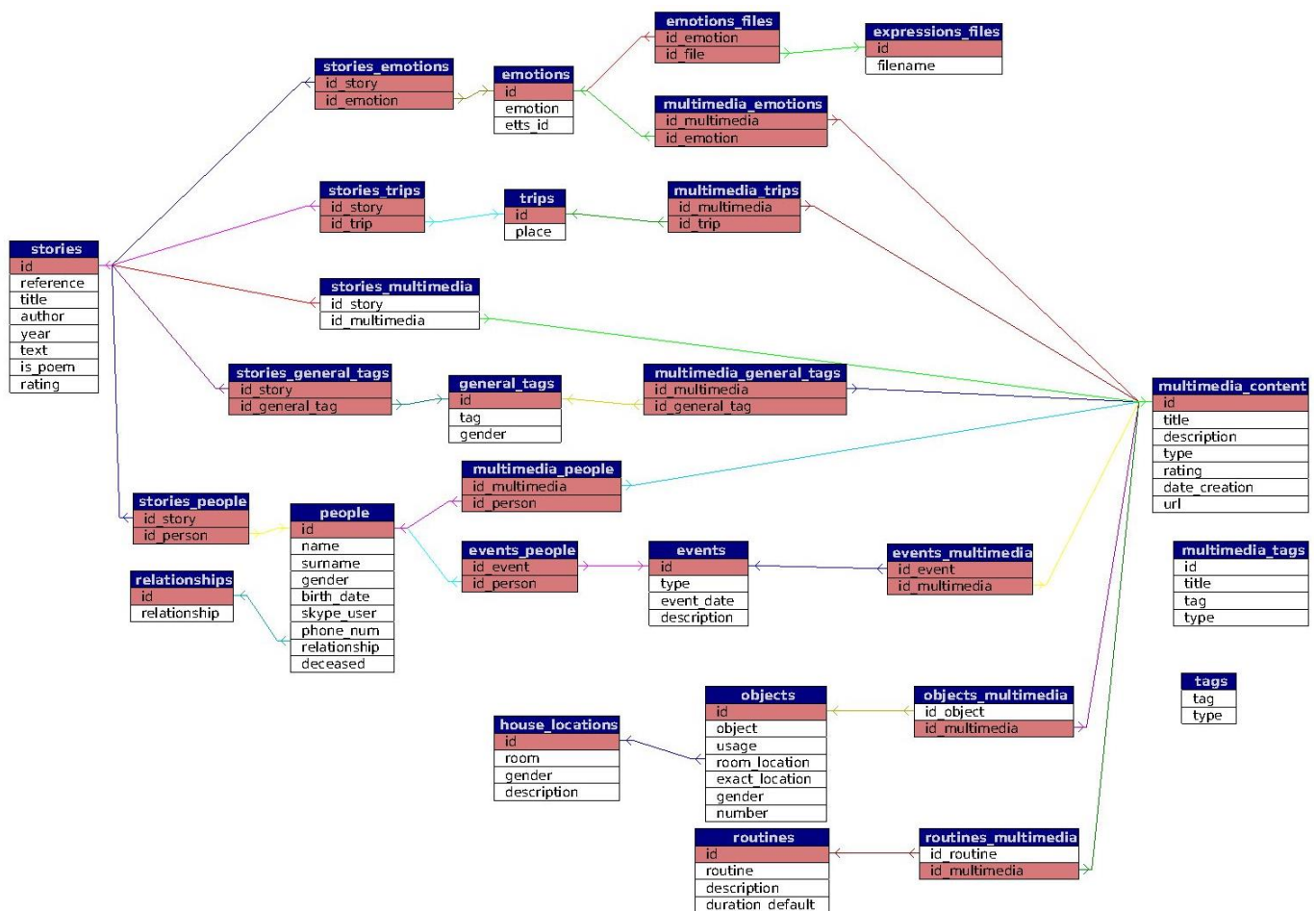


Figura 5.8 - Relación entre las tablas y vistas MySQL de “robalz”

En los siguientes subapartados se detallan los campos y sus correspondientes tipos de las tablas y vistas necesarias. La información relevante para la aplicación se muestra sombreada en cada una de las tablas.

#### ▪ Tabla “multimedia\_content”

La tabla 5.8 contiene la información relacionada con las funcionalidades música, fotos, tv/videos y noticias.

- *title*: contiene el título de cada tarea.
- *description*: contiene la descripción de la tarea.
- *type*: es el tipo de funcionalidad (música → *music*, fotos → *photo*, tv/video → *tv* y *video* y noticias → *news*).
- *url*: dirección en la cual se encuentra el archivo.

multimedia_content		
Columna	Tipo	Ejemplo
<i>id</i>	int(11)	23
<i>title</i>	varchar(255)	Simpsons_bart_maggie
<i>description</i>	text	Bart cogiendo en brazos a Maggie

multimedia_content		
Columna	Tipo	Ejemplo
<i>type</i>	enum('photo', 'video', 'music', 'radio', 'tv', 'news')	<i>photo</i>
<i>rating</i>	double	NULL
<i>date_creation</i>	date	NULL
<i>url</i>	text	/mnt/external_sd/multimedia/images/simpsons/simpsons_bart_maggie.jpg

Tabla 5.8 - Tabla “multimedia\_content”

#### ▪ Tabla “stories”

La tabla 5.9 contiene la información relacionada con la funcionalidad historias.

- *reference*: palabra que sirve para referenciar una historia de forma única.
- *title*: contiene el título de las historias.
- *author*: contiene el nombre del autor de las historias.

stories		
Columna	Tipo	Ejemplo
<i>id</i>	int(11)	2
<i>reference</i>	varchar(255)	Pirata
<i>title</i>	varchar(255)	La canción del pirata
<i>author</i>	varchar(255)	Espronceda
<i>year</i>	year(4)	NULL
<i>text</i>	text	-
<i>is_poem</i>	tinyint(4)	1
<i>rating</i>	double	NULL

Tabla 5.9 - Tabla “stories”

#### ▪ Vista “multimedia\_tags”

La tabla 5.10 contiene la información relacionada con la funcionalidad multimedia. Esta vista almacena las etiquetas que al menos contienen un archivo multimedia.

- *title*: contiene el título de la tarea multimedia.
- *tag*: es la etiqueta relacionada con la tarea.
- *type*: es el tipo de etiqueta.

Un mismo título puede tener varias etiquetas y ser de distintos tipos.

multimedia_tags			
Columna	Tipo	Ejemplos	
<i>id</i>	int(11)	20	20
<i>title</i>	varchar(255)	Simpson_marge_homer	Simpson_marge_homer
<i>tag</i>	varchar(255)	Marge	<i>happy</i>
<i>type</i>	varchar(12)	<i>person</i>	<i>emotion</i>

Tabla 5.10 - Vista “multimedia\_tags”

#### ▪ Vista “tags”

La tabla 5.11 contiene la información relacionada con la funcionalidad multimedia. Esta vista almacena el conjunto de todas las etiquetas.

- *tag*: es la etiqueta relacionada con la tarea.
- *type*: es el tipo de etiqueta.

tags			
Columna	Tipo	Ejemplos	
<i>tag</i>	varchar(255)	Marge	<i>happy</i>
<i>type</i>	varchar(12)	<i>person</i>	<i>emotion</i>

Tabla 5.11 - Vista “tags”

#### ▪ Vista “teleop\_mobile”

Al seleccionar cada una de las funcionalidades de la aplicación se debe mostrar una lista con las tareas que hay disponibles, según se detalla en el requisito RF-05 (Seleccionar funcionalidad). Debido a que la información se encuentra almacenada en tablas y vistas distintas, se ha considerado necesario crear una nueva vista denominada “teleop\_mobile” que almacene todo el contenido relevante para la aplicación de todas las funcionalidades que se van a implementar.

La tabla 5.12 contiene la información necesaria de todas las funcionalidades requeridas en la aplicación (música, fotos, tv/videos, noticias, historias y multimedia).

- *title*: contiene el título de cada tarea.
- *description*: contiene la descripción o el autor de la tarea.
- *type*: es el tipo de funcionalidad (música → *music*, fotos → *photo*, tv/video → *tv* y *video*, noticias → *news*, historias → *story* y multimedia → *tags*).
- *url*: dirección en la cual se encuentra el archivo de música, fotos, tv/videos y noticias.
- *reference*: palabra que sirve para referenciar una historia de forma única.

teleop_mobile					
Columna	Tipo	Ejemplos			
<i>id</i>	int(11)	2	5	22	27
<i>title</i>	varchar(255)	La canción del pirata	Antena 3	simpsons_bart_lisa	Homer
<i>description</i>	text	Espronceda	Antena 3	foto en la que Lisa cura una herida a Bart	<i>person</i>
<i>type</i>	varchar(5)	<i>story</i>	<i>tv</i>	<i>photo</i>	<i>tags</i>
<i>url</i>	text	NULL	http://antena3-aos1-apple-live.adaptive.level3.net/apple/antena3/channel01/antena_3_hd_940K_768x432_baseline.m3u8	/mnt/external_sd/multimedia/images/simpsons/simpsons_bart_lisa.jpg	NULL
<i>reference</i>	varchar(255)	pirata	NULL	NULL	NULL

Tabla 5.12 - Vista “teleop\_mobile”

Para crear esta vista son necesarias las tablas “multimedia\_content” y “stories” y las vistas “multimedia\_tags” y “tags”. En la figura 5.9 se muestra el código para la creación de la vista.

```
CREATE VIEW `teleop_mobile` AS SELECT `s`.`id` AS `id`,`s`.`title` AS `title`,`s`.`author` AS `description`,`story` AS `type`,`NULL` AS `url`,`s`.`reference` AS `reference` FROM `stories` `s` UNION SELECT `m`.`id` AS `id`,`m`.`title` AS `title`,`m`.`description` AS `description`,`m`.`type` AS `type`,`m`.`url` AS `url`,`NULL` AS `reference` FROM `multimedia_content` `m` UNION SELECT `mt`.`id` AS `id`,`mt`.`tag` AS `title`,`mt`.`type` AS `description`,`tags` AS `type`,`NULL` AS `url`,`NULL` AS `reference` FROM ((`multimedia_tags` `mt` JOIN `tags` `t` ON((`t`.`tag` = `mt`.`tag`))));
```

Figura 5.9 - Código creación vista “teleop\_mobile”

En la tabla 5.13 se muestra la equivalencia de cada uno de los campos de las tablas y vistas empleadas para llevar a cabo la creación de la vista “teleop\_mobile”.

teleop_mobile	<i>id</i>	<i>title</i>	<i>description</i>	<i>type</i>	<i>url</i>	<i>reference</i>
stories	<i>id</i>	<i>title</i>	<i>author</i>	“story”	NULL	<i>reference</i>
mutlimedia_content	<i>id</i>	<i>title</i>	<i>description</i>	<i>type</i>	<i>url</i>	NULL
multimedia_tags	<i>id</i>	<i>tag</i>	<i>type</i>	“tags”	NULL	NULL

Tabla 5.13 - Matriz de equivalencia de tablas

### ▪ Tabla “state\_content”

Para consultar el estado del robot según se define en el requisito RF-03 (Consultar estado del robot) es necesario almacenar la información correspondiente a la última tarea que el usuario envía al robot. Para ello se ha creado una tabla denominada “state\_content” que tendrá tres campos: id, título y tipo.

El contenido de esta tabla también permitirá cumplir con los requisitos de parar la tarea que se está realizando (RF-08 Parar tarea), pausarla (RF-09 Pausar tarea) y reanudarla (RF-10 Reanudar tarea).

La tabla 5.14 contiene la información de la última tarea enviada al robot.

- *title*: almacena la información relevante de cada funcionalidad: la url de las funcionalidades música, fotos, tv/video y noticias, la referencia (*reference*) de la funcionalidad historias y el título (*title*) de la funcionalidad multimedia.
- *type*: es el tipo de funcionalidad (música → *music*, fotos → *photo*, tv/video → *tv* y *video*, noticias → *news*, historias → *story* y multimedia → *tags*).

state_content					
Columna	Tipo	Ejemplos			
<i>id</i>	int(11)	1	2	3	4
<i>title</i>	varchar(255)	pirata	http://antena3-aos1-apple-live.adaptive.level3.net/apple/antena3/channel01/antena_3_hd_940K_768x432_baseline.m3u8	/mnt/external_sd/multimedia/images/simpsons/simpsons_bart_lisa.jpg	Homer
<i>type</i>	varchar(255)	story	tv	photo	tags

Tabla 5.14 - Tabla “state\_content”

En la figura 5.10 se muestra el código para la creación de la tabla.

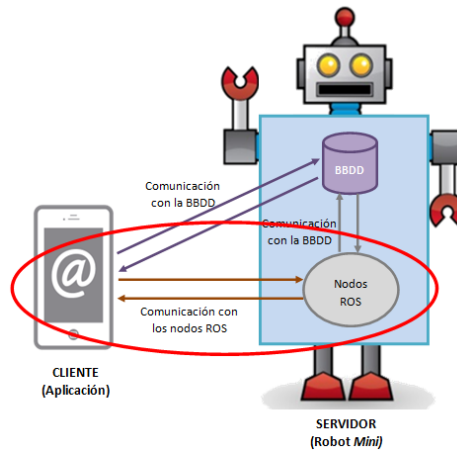
```
CREATE TABLE IF NOT EXISTS `state_content` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `title` varchar(255) COLLATE utf8_spanish_ci NOT NULL,
  `type` varchar(255) COLLATE utf8_spanish_ci NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci
AUTO_INCREMENT=1 ;
```

Figura 5.10 - Código creación tabla “state\_content”



### 5.3.2 Aplicación – ROS

En este subapartado se detallan los elementos que componen la comunicación entre la aplicación y ROS (sección remarcada de la figura 5.11).



*Figura 5.11 - Esquema general: Aplicación – ROS*

La aplicación también debe comunicarse con la arquitectura ROS del robot para que el usuario pueda solicitar al robot realizar, parar, pausar y reanudar una tarea y consultar el estado de éste. Teniendo en cuenta las características de comunicación entre las aplicaciones web y ROS del apartado 5.1.1.2, y los requisitos no funcionales de comunicación, almacenamiento y desempeño, los elementos que permiten realizar dicha comunicación se muestran en la figura 5.12.

Para comprender mejor la figura, se explica el proceso de comunicación entre la aplicación y ROS al seleccionar una canción en la aplicación (CU-06: Seleccionar tarea).

- El usuario selecciona una canción de la lista de canciones disponibles para que el robot la reproduzca.
- La aplicación conecta con el servidor *rosbridge* del robot mediante *WebSockets* utilizando la librería JavaScript *roslibjs*.
- La aplicación envía un mensaje ROS al nodo “*activity\_persistence\_module*”.
- El robot reproduce la canción seleccionada por el usuario.
- Se actualiza el estado del robot con la canción que está reproduciendo.

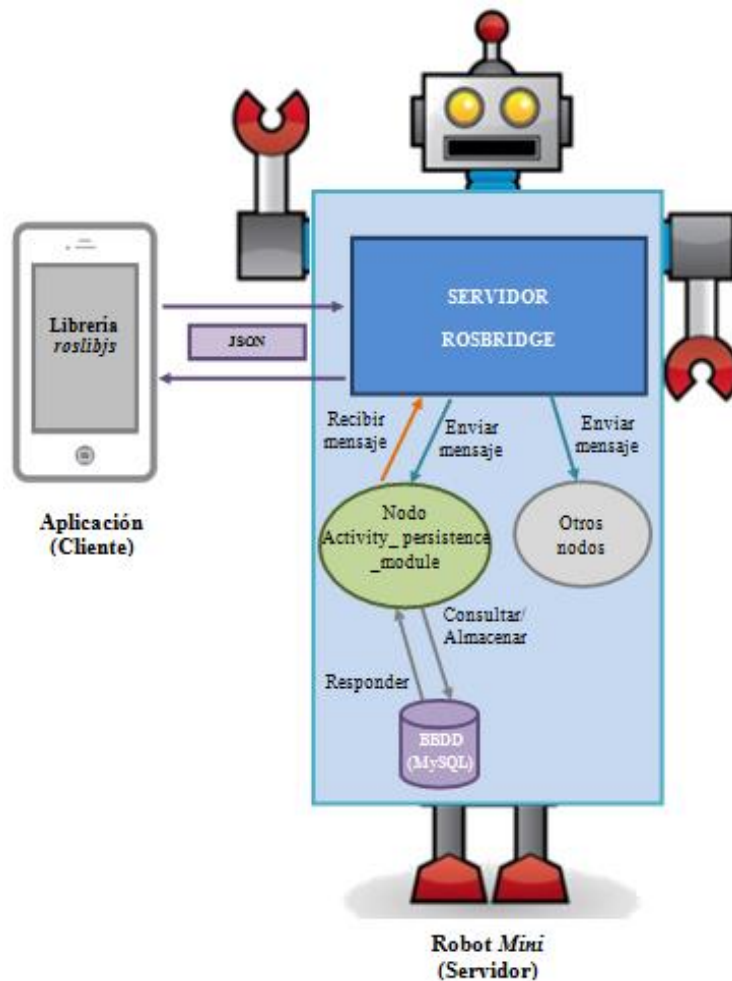


Figura 5.12 - Esquema de comunicación: Aplicación – ROS

En la figura 5.13 se muestra el diagrama de secuencia para el ejemplo descrito:

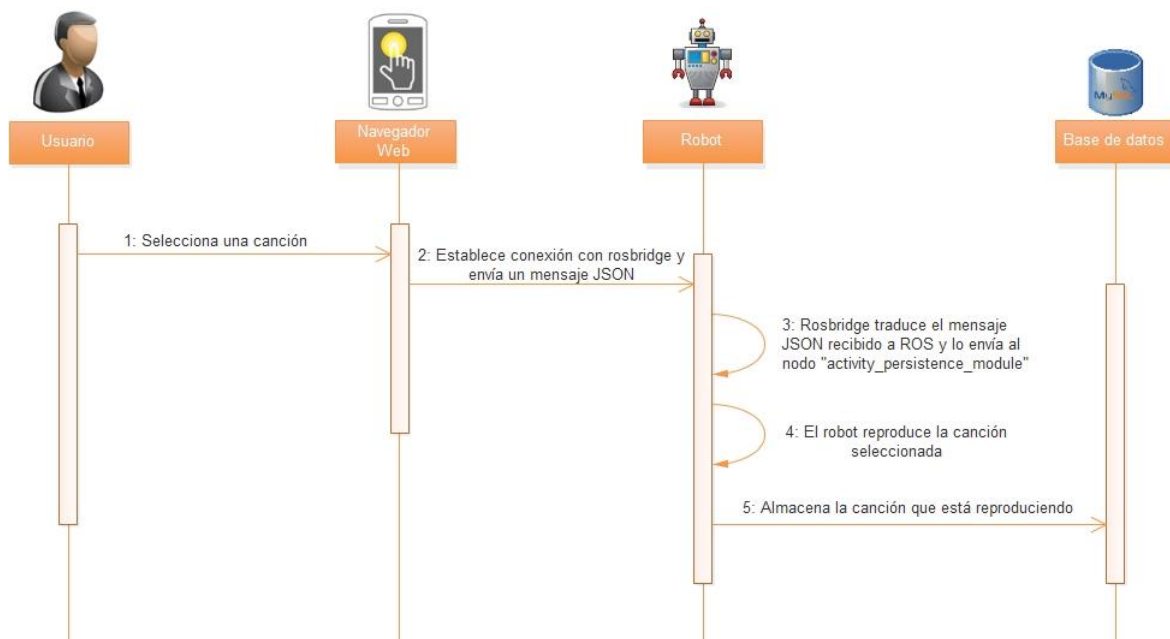


Figura 5.13 - Diagrama de secuencia: Aplicación – ROS

### 5.3.2.1 Rosbridge

En este subapartado se detalla el proceso de comunicación entre la aplicación y el servidor *rosbridge* (sección remarcada de la figura 5.14).

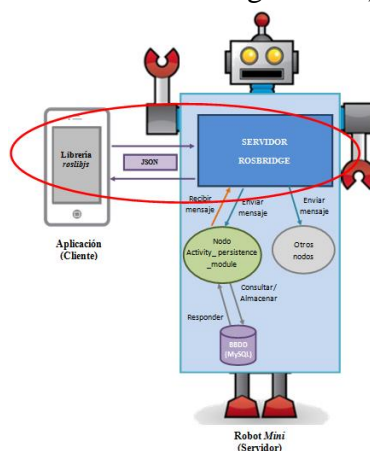


Figura 5.14 - Esquema de comunicación: Rosbridge

*Rosbridge* contiene un servidor (*rosbridge\_server*) que se ejecuta dentro del entorno de ROS. Este servidor proporciona una capa de transporte *WebSocket*, es decir, una capa de comunicación bidireccional de baja latencia entre cliente y servidor. Al proporcionar una conexión *WebSocket*, el servidor de *rosbridge* permite a las páginas web interactuar con ROS usando el protocolo *Rosbridge* [33]. En el lado del cliente, se hace uso de la librería JavaScript *roslibjs* para llevar a cabo dicha interacción entre el cliente y el servidor mediante *WebSockets*.

El servidor de *rosbridge* crea una conexión *WebSocket* e intercambia mensajes basados en JSON con los clientes conectados. La estructura de los mensajes JSON intercambiados entre los clientes y el servidor de *rosbridge* se define en el protocolo *Rosbridge*. Para realizar el intercambio de mensajes entre el cliente y el servidor se hace uso de la librería *rosbridge\_library*. El servidor de *rosbridge* pasa cualquier mensaje JSON del *WebSocket* a la *rosbridge\_library*, que se encarga de convertir las cadenas JSON en una llamada de ROS. También realiza el proceso contrario, es decir, convierte las respuestas de ROS en JSON, las pasa al servidor de *rosbridge*, y éste las envía a través del *WebSocket* al cliente.

Por otro lado, *rosbridge* intercambia mensajes ROS con los distintos nodos ROS del robot. Para poder controlar qué tareas están en ejecución, actualizar el estado, etc., se ha diseñado e implementado un nodo ROS denominado “*activity\_persistence\_module*” que se detalla en el siguiente apartado. En el futuro, dicho nodo llegará a formar parte del control central del robot porque actualmente éste está en desarrollo.

### 5.3.2.2 Nodos ROS

En este subapartado se describen los nodos ROS (sección remarcada de la figura 5.15) que utiliza la aplicación para comunicarse con el robot.

Como se ha descrito en el apartado 2.4.1, los nodos son procesos encargados de una función específica y se comunican entre sí mediante paso de mensajes.

En este proyecto se va a hacer uso de los dos tipos de comunicación entre nodos, la síncrona y la asíncrona. Los nodos encargados de parar, pausar y reanudar una tarea estaban implementados en el robot utilizando la comunicación asíncrona antes de la realización de este proyecto. Sin embargo para la realización de tareas y la consulta del estado del robot se ha decidido utilizar la comunicación síncrona, de manera que la respuesta recibida se puede mostrar al usuario.

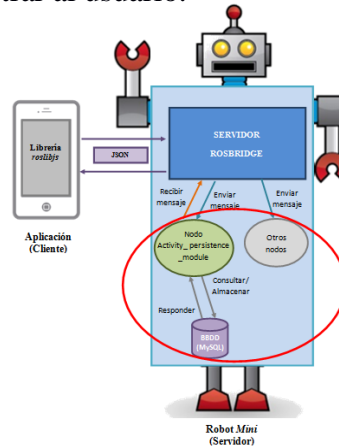


Figura 5.15 - Esquema de comunicación: Nodos ROS

#### ▪ Comunicación síncrona

Para la comunicación síncrona se ha realizado el diseño de dos servicios, uno para la realización de las tareas “/alz/request\_activity” y otro para consultar el estado del robot “/alz/current\_activity”.

En la figura 5.16 se muestra pseudocódigo del servicio “request\_activity”:

```

Establecer conexión con la base de datos "robalz"
Consultar la tabla "state_content"

SI hay datos ENTONCES
    SI req.titulo == titulo_tarea_state_content ENTONCES
        Escribir "Es la misma tarea"
    SI NO
        Parar la tarea de la tabla "state_content" publicando el mensaje
        de parar en el tema correspondiente según el tipo de la tarea de la
        tabla "state_content".
        Eliminar la tarea de la tabla "state_content"
        Insertar la tarea pedida por el usuario en la tabla
        "state_content"
        Publicar el mensaje de la tarea pedida en el tema
        correspondiente según el tipo de la tarea pedida.
    FIN SI

SI NO
    Escribir "No se ha encontrado contenido en state_content"
    Insertar la tarea pedida por el usuario en la tabla "state_content"
    Publicar el mensaje de la tarea pedida en el tema correspondiente
    según el tipo de la tarea pedida.

FIN-SI
  
```

Figura 5.16 - Pseudocódigo del servicio “request\_activity”

En la figura 5.17 se muestra pseudocódigo del servicio “current\_activity”:

```

Establecer conexión con la base de datos "robalz"
Consultar la tabla "state_content"

SI hay datos ENTONCES

    Consultar la vista "teleop_mobile" para obtener el título y tipo
    de la tarea almacenada en "state_content".

    //Devolver el tipo y título de la tarea que está realizando el
    robot
    res.actividad = tipo_tarea_teleop_mobile
    res.titulo = titulo_tarea_teleop_mobile

SI NO
    //El robot no está realizando ninguna tarea
    res.actividad = ""
    res.titulo = ""
FIN-SI

```

*Figura 5.17 - Pseudocódigo del servicio “current\_activity”*

#### ▪ Comunicación asíncrona

Para la comunicación asíncrona, actualmente las funcionalidades música, fotos, tv/videos y noticias únicamente disponen de la posibilidad de parar tarea. Para ello se envía un mensaje de tipo “tv\_on\_demand/multimediaPlayer” en el tema “/alz/tablet\_player” con contenido “back” en todos los campos del mensaje. Sin embargo, las funcionalidades historias y multimedia sí disponen de reanudar, pausar y parar tareas:

- Para la funcionalidad historias se envía un mensaje de tipo “std\_msgs/Int16” con contenido “1” en el tema “/alz/STORY\_TELLING\_START\_STOP” para pausar o reanudar historias y con contenido “0” en el tema “/alz/STORY\_TELLING\_STOP” para pararlas.
- Para la funcionalidad multimedia se envía un mensaje de tipo “multimedia\_player/MultimTagMsg” en el tema “/alz/MULTIMEDIA\_PLAYER” con contenido “stop”, “pause” o “resume” si se para, pausa o reanuda una tarea respectivamente.

Además al parar una tarea es necesario otro tema que se encargue de eliminarla de la tabla “state\_content” para actualizar el estado del robot. Este tema se denomina “/alz/activity\_finished”, el tipo del mensaje es “std\_msgs/String” y su contenido es el tipo de la tarea a parar.

En la figura 5.18 se muestra el pseudocódigo del tema “activity\_finished” diseñado:

```

Establece conexión con la base de datos "robalz"
Consultar la tabla "state_content"

SI hay datos ENTONCES
    SI tipo_tarea == tipo_tarea_state_content ENTONCES
        Eliminar la tarea de la tabla state_content

    SI NO
        Escribir "No se elimina porque no es la misma actividad"
    FIN SI
SI NO
    escribir "No se ha encontrado contenido en state_content"

FIN-SI

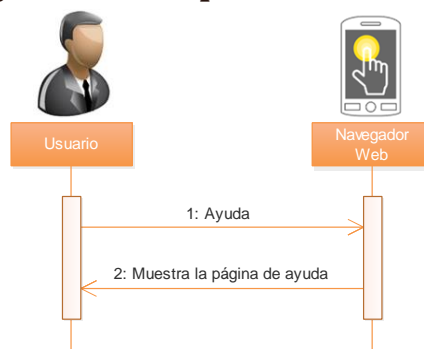
```

*Figura 5.18 - Pseudocódigo del tema "activity\_finished"*

## 5.4 Diagramas de secuencia

Después de describir cada elemento del sistema y a fin de entender su funcionamiento, se completa este capítulo con un diagrama de secuencia por cada caso de uso definido en el capítulo 3.

### 5.4.1 Consultar ayuda de la aplicación



*Figura 5.19 - Diagrama de secuencia: consultar ayuda de la aplicación*

### 5.4.2 Cambiar idioma de la aplicación



*Figura 5.20 - Diagrama de secuencia: cambiar idioma de la aplicación*

### 5.4.3 Consultar estado del robot

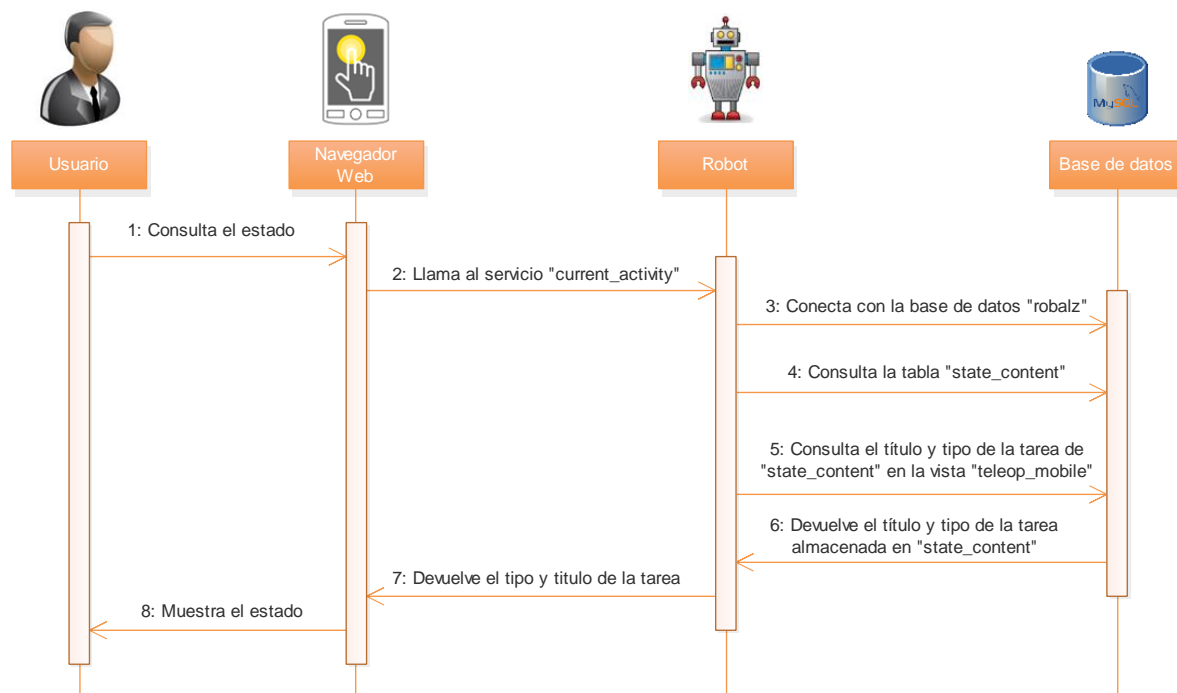


Figura 5.21 - Diagrama de secuencia: consultar estado del robot

### 5.4.4 Buscar tarea

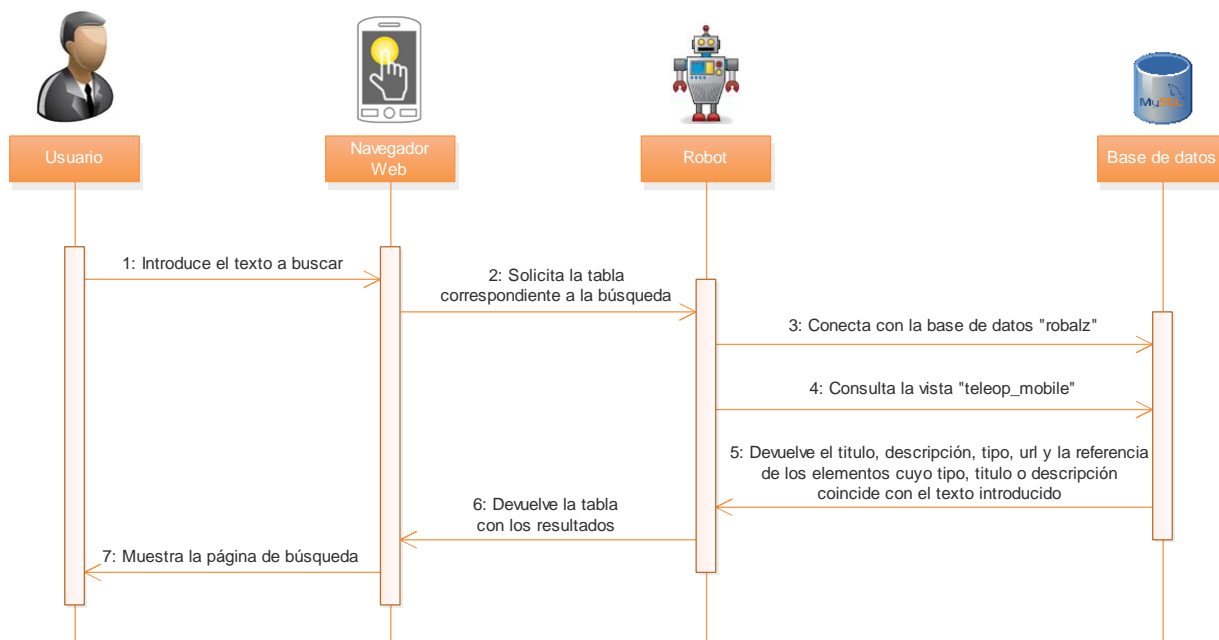


Figura 5.22 - Diagrama de secuencia: buscar tarea

### 5.4.5 Seleccionar funcionalidad

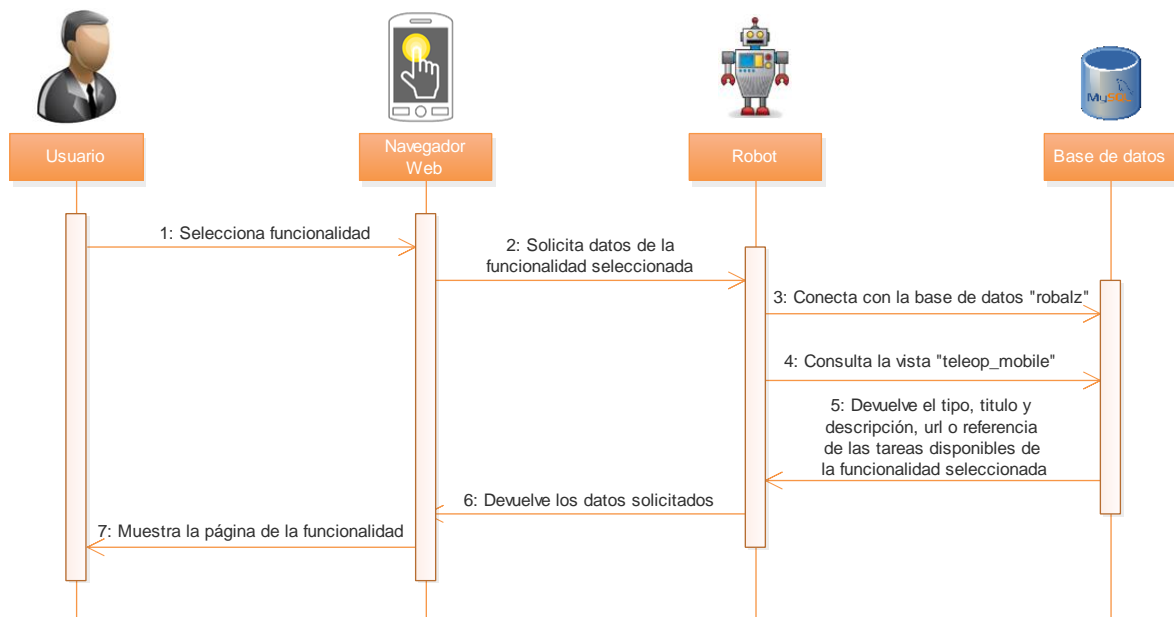


Figura 5.23 - Diagrama de secuencia: seleccionar funcionalidad

### 5.4.6 Seleccionar tarea



Figura 5.24 - Diagrama de secuencia: seleccionar tarea



### 5.4.7 Realizar tarea

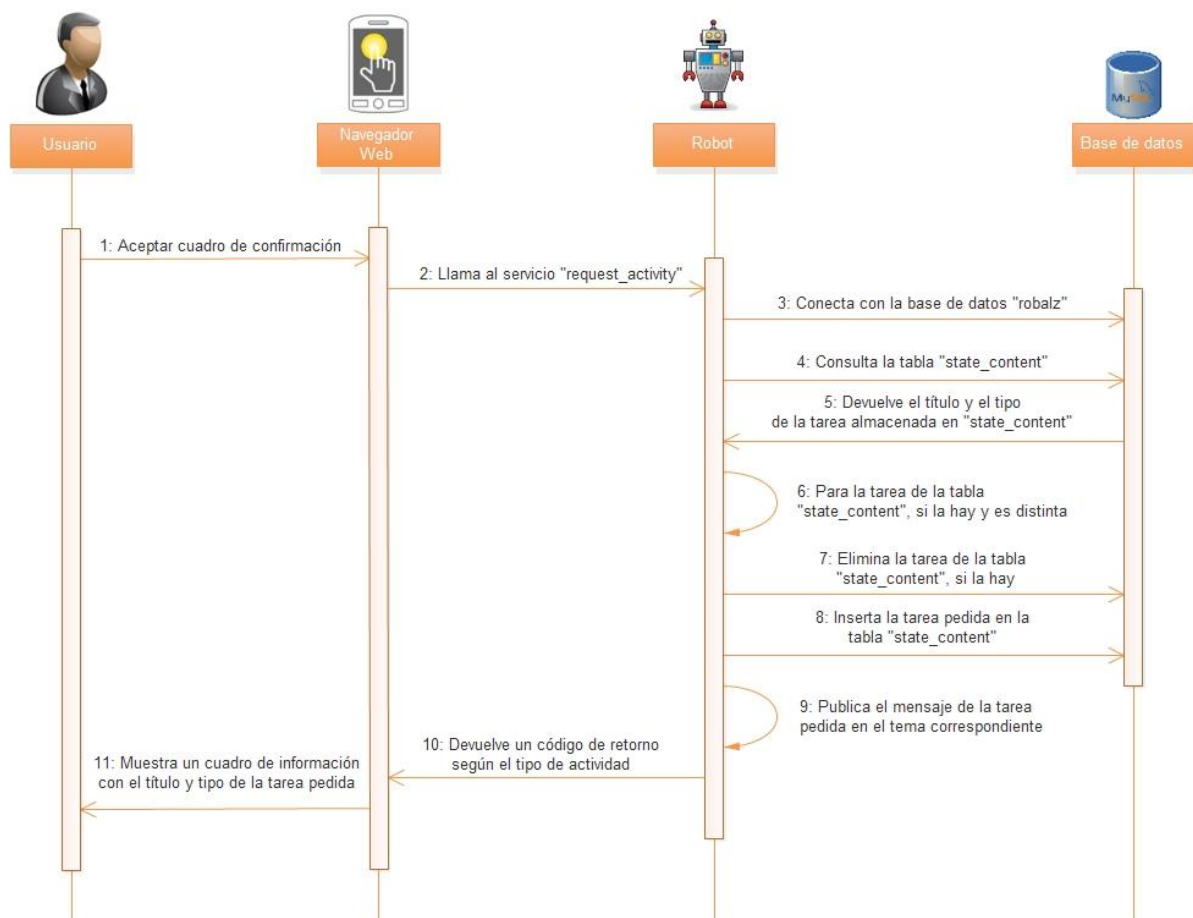


Figura 5.25 - Diagrama de secuencia: realizar tarea

### 5.4.8 Pausar/reanudar tarea

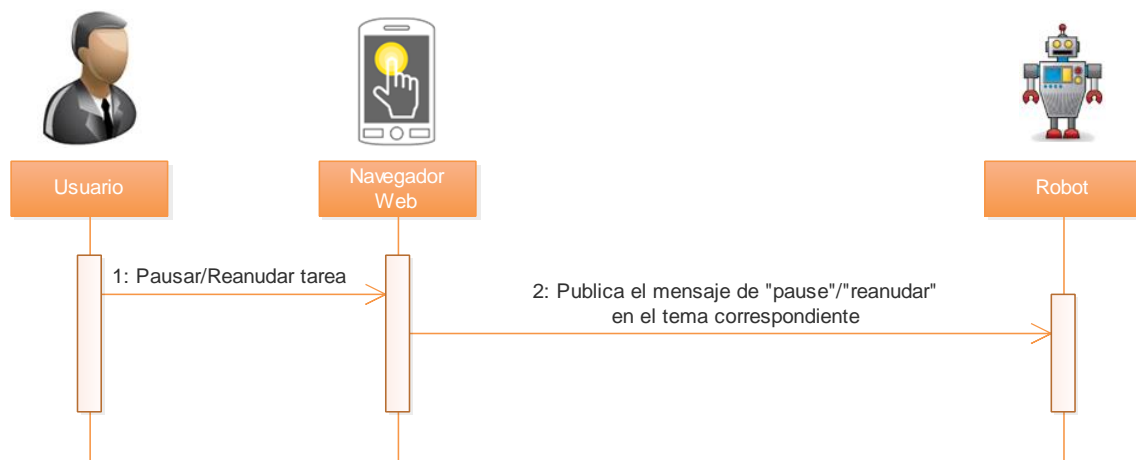


Figura 5.26 - Diagrama de secuencia: pausar/reanudar tarea

### 5.4.9 Parar tarea

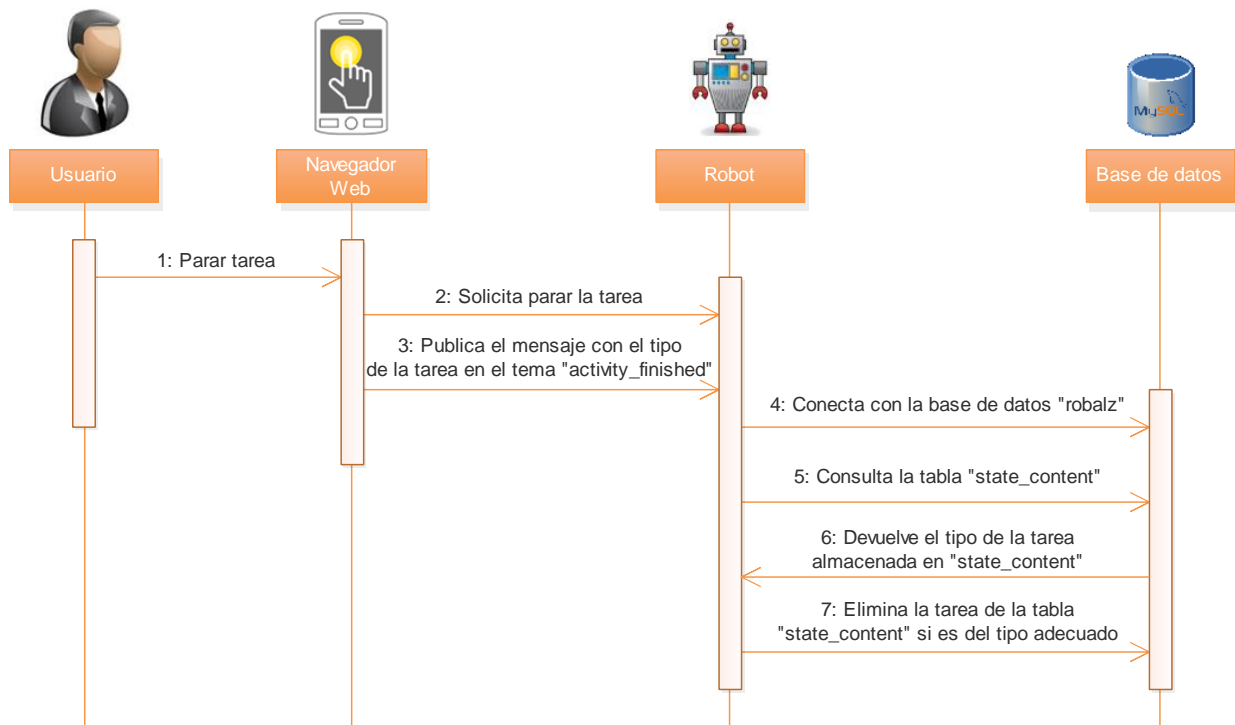
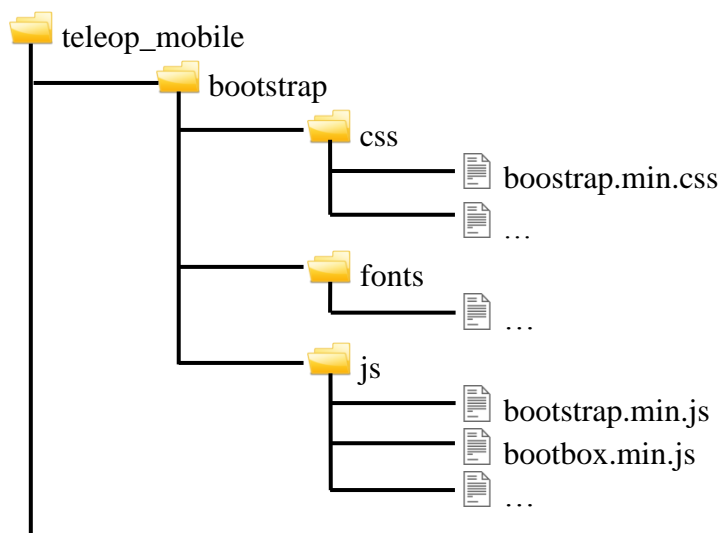


Figura 5.27 - Diagrama de secuencia: parar tarea

## 5.5 Estructura de ficheros aplicación v.1.0

Los ficheros de la primera versión de la aplicación web deben estar en el lado del servidor debido a que todas las páginas están implementadas con código PHP. La estructura de ficheros (figura 5.28) debe estar almacenada en la carpeta raíz del servidor web. En el caso de Apache2.2 la carpeta raíz por defecto es /var/www, por lo que en dicho directorio es donde deben guardarse los siguientes directorios y ficheros:



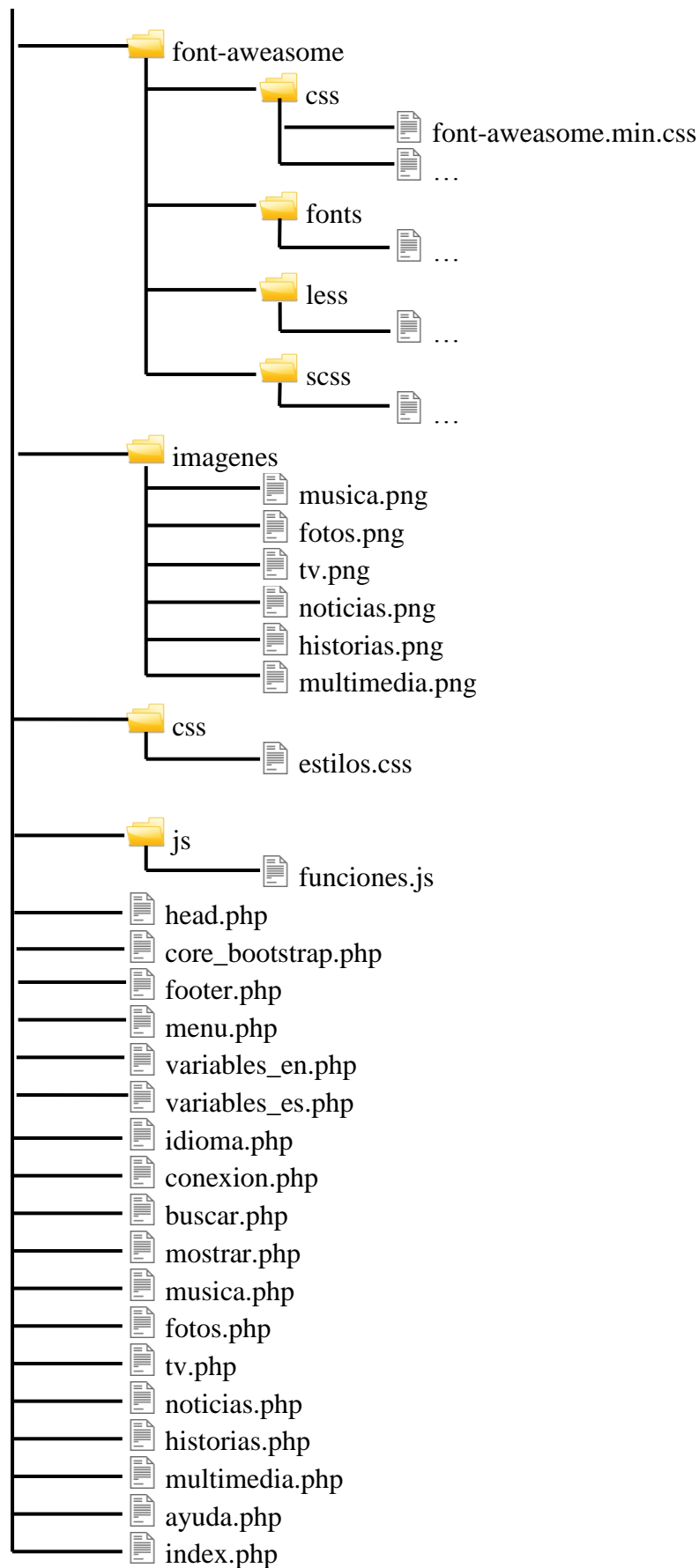


Figura 5.28 - Estructura de ficheros de la aplicación web v.1.0

Los ficheros PHP se pueden dividir en cuatro grupos:

- 1) Ficheros para cada página de la interfaz:
  - index.php: muestra la página principal.
  - musica.php: muestra la página de la funcionalidad música.
  - fotos.php: muestra la página de la funcionalidad fotos.
  - tv.php: muestra la página de la funcionalidad tv/videos.
  - noticias.php: muestra la página de la funcionalidad noticias.
  - historias.php: muestra la página de la funcionalidad historias.
  - multimedia.php: muestra la página de la funcionalidad multimedia.
  - ayuda.php: muestra la página de la ayuda de la interfaz.
- 2) Ficheros comunes a cada página de la interfaz:
  - head.php: contiene todos los elementos de la cabecera HTML.
  - core\_bootstrap.php: incluye los *scripts* del *framework* Bootstrap, de *bootbox* para los cuadros de diálogo y de la librería jQuery.
  - footer.php: contiene el texto que se muestra en la parte inferior de la página.
  - menu.php: contiene los elementos que conforman la barra de menú de la interfaz.
- 3) Ficheros de selección de idioma:
  - variables\_es.php: contiene el texto en español que se muestra en cada página de la interfaz.
  - variables\_en.php: contiene el texto en inglés que se muestra en cada página de la interfaz.
  - idioma.php: permite cargar el fichero variables\_es.php o variables\_en.php en función del idioma escogido por el usuario.
- 4) Contenido de las tablas de cada funcionalidad:
  - conexion.php: realiza la conexión a la base de datos MySQL del robot.
  - buscar.php: realiza la búsqueda del texto introducido por el usuario en la base de datos.
  - mostrar.php: representa en una tabla las coincidencias encontradas tras la búsqueda.

Se ha creado el fichero estilos.css para cambiar la apariencia de algunos elementos HTML de la interfaz, y el fichero funciones.js que contiene todas las funciones JavaScript que hacen posible mostrar los cuadros de confirmación e información, realizar la conexión con el servidor de *rosbridge*, llamar a los servicios ROS y publicar mensajes en los temas ROS del robot.

## 5.6 Capturas de la aplicación v.1.0

En este apartado se muestran algunas de las páginas que forman parte de la primera versión de la aplicación.

### 5.6.1 Página principal y menú

En la imagen de la izquierda de la figura 5.29 se muestra la página principal de la aplicación web v.1.0. Para acceder a ella, hay que utilizar el navegador web del dispositivo móvil e introducir la URL “163.117.150.198/teleop\_mobile”. El número 163.117.150.198 hace referencia a la IP del robot o del router con dirección pública que le dé acceso. Se debe modificar dicha dirección si cambia o se le asigna un dominio al robot.

En la página principal se tiene acceso al menú de la aplicación (parte superior roja), a las diferentes funcionalidades de entretenimiento que implementa mediante botones y a la consulta del estado del robot (cuadro verde de la parte inferior).

Por otro lado, en la imagen de la derecha de la figura 5.29, se muestra el menú de la aplicación que es común a todas las páginas de ésta. Por defecto se encuentra plegado, teniendo siempre visible el enlace *Inicio/Estado* para acceder a la página principal, y oculto el resto de funcionalidades de entretenimiento, la barra de búsqueda, la selección de idioma y la consulta de la guía para usar la aplicación.

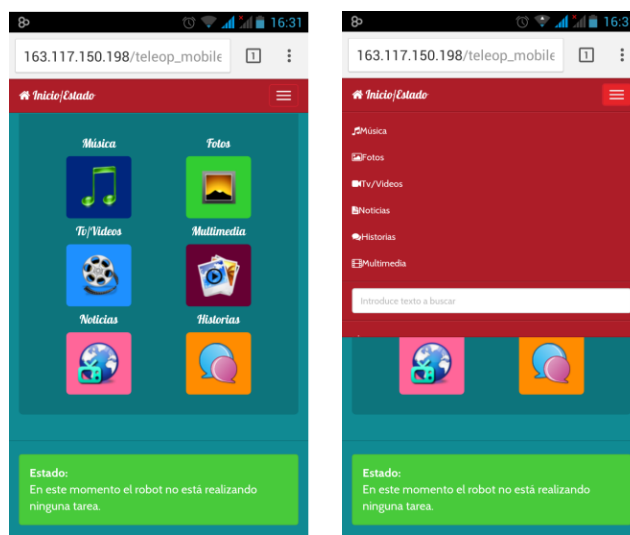


Figura 5.29 - Web v.1.0: Página principal (izq.) y Menú desplegado (dcha.)

### 5.6.2 Páginas de las funcionalidades

En la imagen de la izquierda de la figura 5.30 se muestra la página de música en la que se listan las canciones disponibles que puede reproducir el robot en la *tablet* que incorpora. Esta funcionalidad actualmente sólo dispone de la opción “parar” ajustándose a la implementación del robot previa al desarrollo del proyecto. Las páginas de las funcionalidades fotos, tv/videos y noticias son similares.

La página de historias difiere de las demás en que el robot cuenta la historia seleccionada, es decir, no hace uso de la *tablet* y además añade las opciones “reanudar” y “pausar” historia.

En la imagen de la derecha de la figura 5.30 se muestra la página de contenidos multimedia que incorpora a las opciones de la página de historias, la selección múltiple. En esta página el usuario puede seleccionar una o varias etiquetas. Una etiqueta es el nombre asignado a un conjunto de contenidos multimedia que tienen en algo en común.

En las tablas de cada funcionalidad se puede ordenar alfabéticamente por “título” o “descripción” pulsando los iconos ▼ (orden de A-Z) y ▲ (orden de Z-A) de la columna que se desea ordenar.

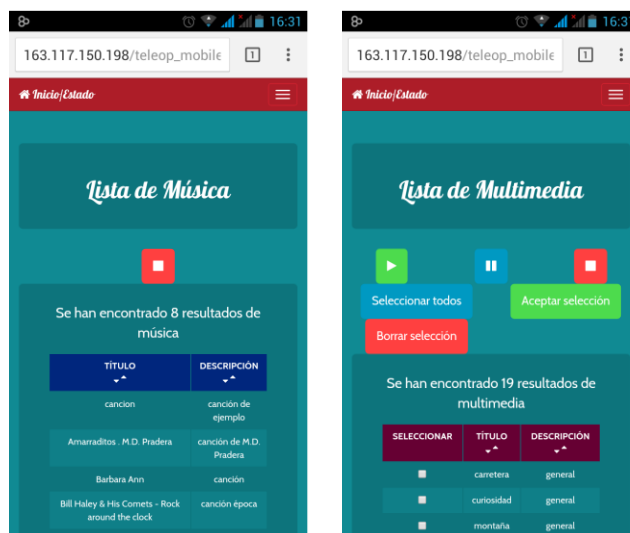


Figura 5.30 - Web v.1.0: Página de música (izq.) y Página de multimedia (dcha.)

### 5.6.3 Página de ayuda y resultados de búsqueda

En la imagen de la izquierda de la figura 5.31 se muestra la página de ayuda que explica los pasos para utilizar la aplicación. Y en la imagen de la derecha se indica dónde hay que introducir el texto de la tarea que se desea buscar. En la búsqueda realizada en la imagen de la izquierda de la figura 5.33, se muestra que se ignoran las tildes, al igual que las mayúsculas y minúsculas.

Se distinguen entre dos tipos de búsqueda: búsqueda en todas las funcionalidades y búsqueda en una funcionalidad. La primera se muestra en la imagen de la derecha de la figura 5.32 y se debe realizar desde cualquiera de las páginas que no sean una funcionalidad de entretenimiento, es decir, desde la página principal o desde la página de ayuda. Sin embargo, si se realiza la búsqueda estando dentro de la página de una funcionalidad, por ejemplo fotos, se realiza la búsqueda del texto introducido que sea una foto (imagen de la izquierda de la figura).

Se puede observar que realizando la búsqueda en todas las funcionalidades se han encontrado dos resultados (una foto y una etiqueta) para la palabra “montaña”. Mientras que realizando la búsqueda en una funcionalidad (en este ejemplo, fotos) únicamente se ha encontrado una foto para la misma palabra.

En la imagen de la derecha de la figura 5.33 se realiza la búsqueda de la palabra “montaña” en la funcionalidad multimedia. En este caso, si se hubiera encontrado más de un resultado se podría hacer una selección múltiple de las coincidencias encontradas.



Figura 5.31 - Web v.1.0: Página de ayuda (izq.) e Insertar búsqueda (dcha.)



Figura 5.32 - Web v.1.0: Búsqueda de fotos (izq.) y Búsqueda general (dcha.)

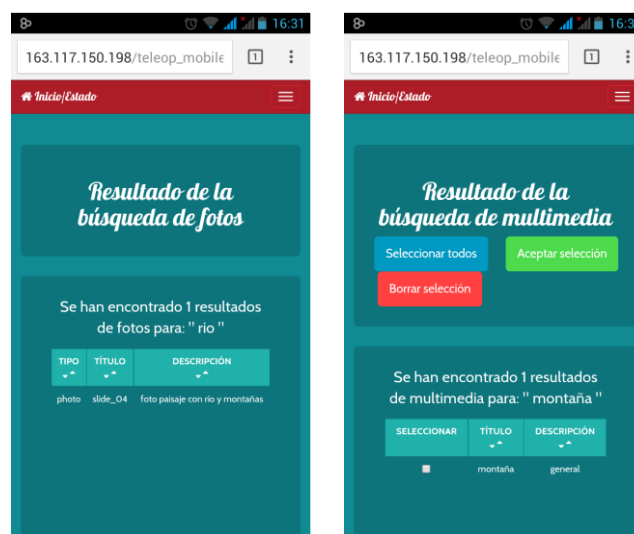


Figura 5.33 - Web v.1.0: Búsqueda de fotos (izq.) y Búsqueda de multimedia (dcha.)

### 5.6.4 Cuadros de confirmación e información

Al seleccionar una de las tareas de las tablas de las funcionalidades, aparece el cuadro de confirmación que se muestra en la imagen de la izquierda de la figura 5.34. Si el usuario pulsa “Aceptar” el robot comenzará a realizar la tarea seleccionada y aparecerá el cuadro de información que se muestra en la imagen derecha de la figura. Mientras que si el usuario pulsa “Cancelar” se cierra el cuadro de confirmación y la aplicación vuelve a la página en la que se encontraba.

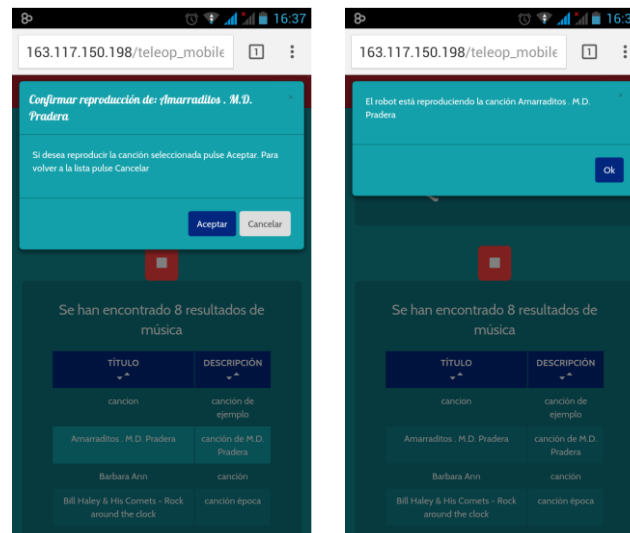


Figura 5.34 - Web v.1.0: Cuadros de Confirmación (izq.) e Información (dcha.)



# Capítulo 6

## Pruebas y evaluación

En este capítulo se detallan las pruebas realizadas para comprobar el correcto funcionamiento de la aplicación y se lleva a cabo un análisis de los resultados obtenidos tras realizar la encuesta del anexo A con usuarios de perfiles distintos. A partir de dichos resultados se quiere conseguir mejorar la interfaz en función de la opinión del usuario, de manera que sea más intuitiva, sencilla y tenga mayor usabilidad para éste.

## 6.1 Pruebas

Esta fase se divide en dos partes, primero se realizan las pruebas unitarias de forma paralela a la fase de implementación para comprobar cada funcionalidad por separado, y a continuación se realizan las pruebas de aceptación para verificar el funcionamiento completo de la aplicación.

Cada una de las pruebas unitarias se ha realizado primero desde el equipo servidor (robot) y después desde un dispositivo móvil conectado a una red distinta a la del robot. Las pruebas unitarias que se han llevado a cabo son las siguientes:

- Comprobar que se realiza la conexión con la base de datos desde la aplicación.
- Probar el servicio “request\_activity”: enviar una canción al robot desde la aplicación y comprobar que éste la realiza y es almacenada en la tabla “state\_content”. Tras verificar su correcto funcionamiento, realizar el mismo proceso con el resto de funcionalidades de entretenimiento.
- Probar el servicio “current\_activity”: comprobar que devuelve el tipo y título de la tarea almacenada en la tabla “state\_content”.
- Probar el tema “activity\_finished”: verificar que al parar una tarea, el robot finaliza la realización de ésta y la elimina de la tabla “state\_content”.
- Probar las opciones de pausar y reanudar las tareas.

Tras finalizar la fase de implementación se llevan a cabo las pruebas de aceptación realizada por un grupo de usuarios con diferentes perfiles, para revisar el correcto funcionamiento de la aplicación completa y asegurar que el sistema desarrollado cumple con los requisitos definidos.

Las pruebas de aceptación a realizar por los usuarios son las siguientes:

- Iniciar la aplicación
- Consultar la ayuda de la aplicación
- Cambiar el idioma de la aplicación
- Consultar el estado del robot
- Probar cada funcionalidad: seleccionar una de ellas (música, fotos, tv/videos, noticias, historias o multimedia), escoger una tarea y confirmar la realización. Comprobar que el robot realiza la tarea seleccionada.
- Consultar el estado del robot para ver que indica la tarea que está realizando.
- Sin parar la tarea con el botón “parar”, seleccionar otra tarea de la misma u otra funcionalidad y comprobar que el robot finaliza la tarea que estaba realizando y comienza a realizar la nueva tarea seleccionada, si ésta no es la misma que la que estaba reproduciendo.
- Consultar de nuevo el estado del robot para comprobar que se ha actualizado con la nueva tarea seleccionada.
- Probar el botón “parar”: seleccionar una tarea y una vez que el robot comience a realizarla, pararla y comprobar que el robot deja de realizar la tarea.
- Comprobar que se ha actualizado el estado del robot.

- Probar los botones “pausar” y “reanudar”: seleccionar una tarea de las funcionalidades historia o multimedia y una vez que el robot comience a realizarla, pausarla y después reanudarla.
- Probar la selección múltiple: seleccionar la funcionalidad multimedia y escoger varias etiquetas para que el robot las realice.
- Realizar una búsqueda de un tipo de funcionalidad.
- Realizar una búsqueda de todas las funcionalidades.

Posteriormente, los mismos usuarios realizan la encuesta del anexo A para valorar los resultados detallados en el siguiente apartado de este capítulo.

## 6.2 Evaluación

En este apartado se detalla cómo se ha llevado a cabo la evaluación de la aplicación v.1.0. Para realizarla, se contó con la participación de dieciséis usuarios que no tenían ninguna relación con el proyecto. A cada usuario se le proporcionó una guía con las pruebas de aceptación que cubren los casos de uso de la aplicación, además de las pruebas que cada uno de ellos hizo voluntariamente. Posteriormente, los usuarios respondieron de manera individual, la encuesta del anexo A que incorpora preguntas acerca de los resultados obtenidos de las pruebas realizadas y de su opinión sobre la apariencia y el funcionamiento de la aplicación.

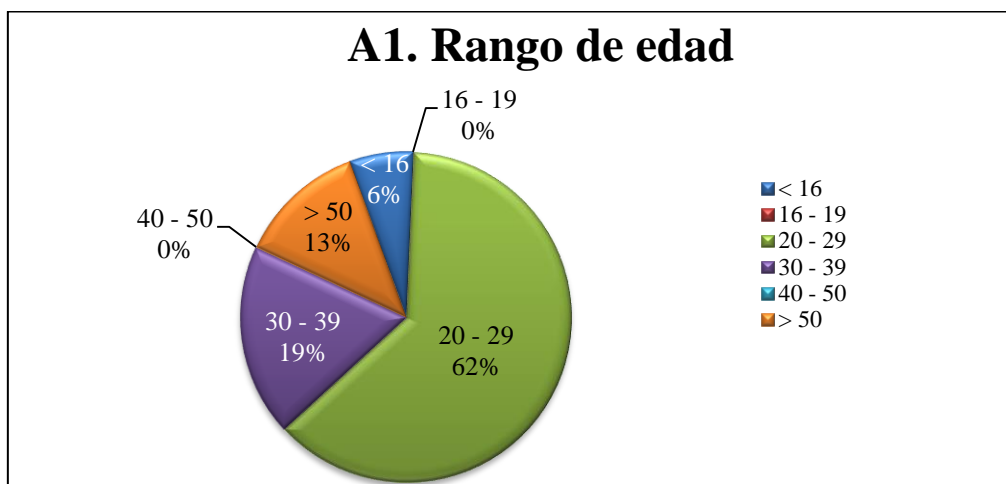
Esta evaluación tiene como objetivo conocer las opiniones y sensaciones de los usuarios al utilizar la aplicación y sus sugerencias para mejorarla. En los siguientes subapartados se muestran los resultados obtenidos representados en gráficas circulares y de barras indicando el porcentaje o el número de usuarios correspondiente.

### 6.2.1 Perfil de usuario

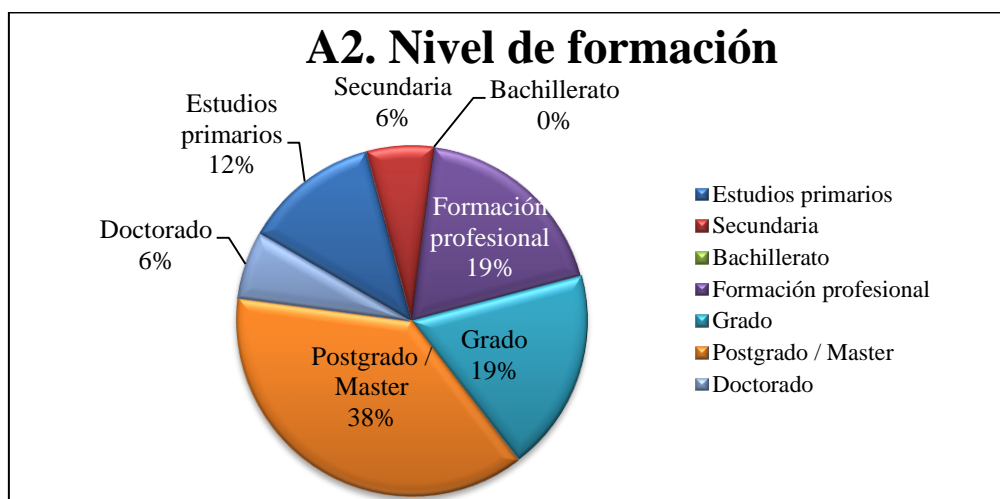
La mayoría de usuarios se encuentran en el rango de edad de 20 - 29 años como se puede apreciar en la gráfica 6.1. No obstante se han probado con otros rangos para verificar que la aplicación es sencilla independientemente de la edad que tenga el usuario.

En la gráfica 6.2 se puede observar que la mayoría de los usuarios tienen un nivel de formación elevado ya que están realizando o tienen un postgrado o máster universitario. Sin embargo, un 12% sólo disponen de estudios primarios y un 6% está cursando la Educación Secundaria Obligatoria (E.S.O), de forma que cualquier usuario de cualquier nivel de formación es capaz de utilizar la aplicación correctamente.

En la gráfica 6.3 se muestra el tipo de formación de los usuarios. Algunos de ellos tienen más de un tipo de formación. Es importante destacar que aunque los usuarios tengan un tipo de formación no tiene por qué estar relacionada con la robótica ni con dispositivos móviles.



Gráfica 6.1 - A1. Rango de edad

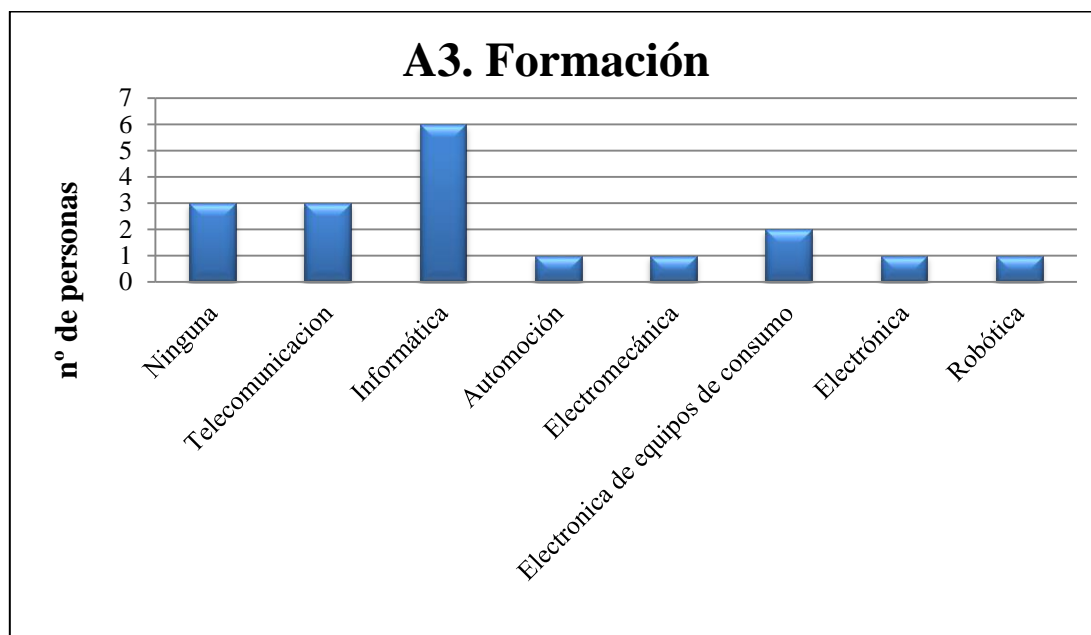


Gráfica 6.2 - A2. Nivel de formación

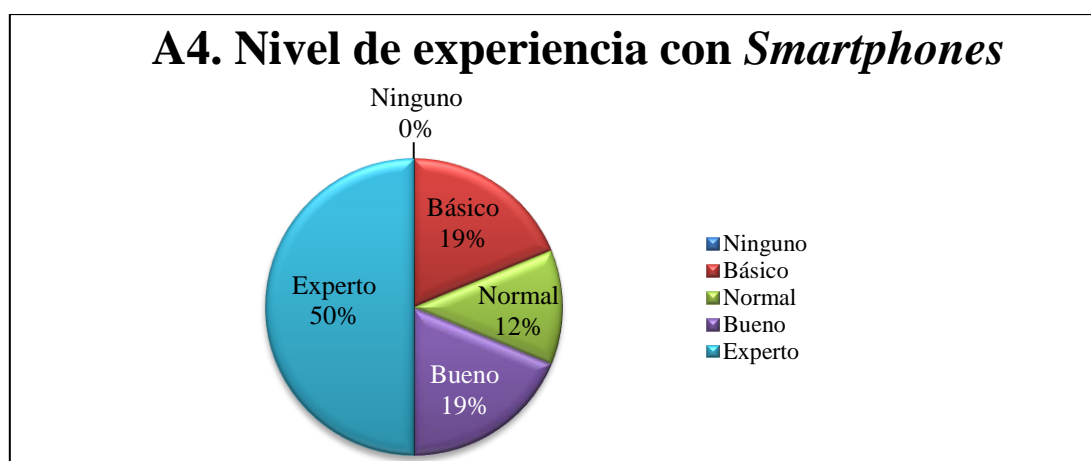
Para el uso de la aplicación es recomendable que el usuario tenga un mínimo de conocimiento para utilizarlo, como por ejemplo ejecutar una aplicación o realizar una búsqueda en Internet. Hoy en día, gran parte de la población tiene al menos un *smartphone* por lo que la mayoría puede utilizar la aplicación de manera satisfactoria.

En la gráfica 6.4 se puede ver que el 19% de los usuarios tienen un nivel básico con dispositivos móviles y han sido capaces de probar todas las funcionalidades que ofrece la aplicación.

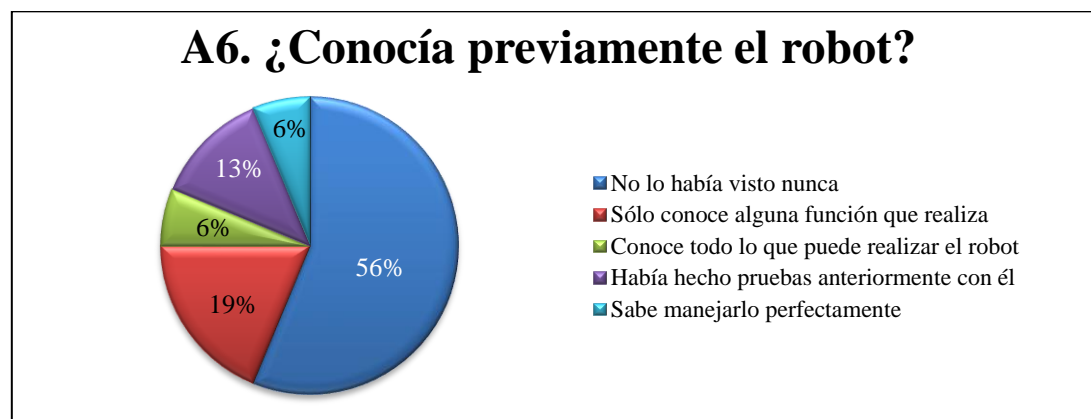
En el apartado A5 de la encuesta se pide al usuario que introduzca el modelo del *smartphone* que ha utilizado para probar la aplicación con el objetivo de confirmar que sea multiplataforma y se pueda usar a una distancia remota con respecto a la posición del robot. En la encuesta se prestó un *smartphone* Android con un tamaño de pantalla de 4.7 pulgadas utilizando la conexión WiFi de la Universidad, pero algunos de los usuarios optaron por emplear sus dispositivos móviles de tamaños de pantalla que comprenden el rango de 4 – 5.7 pulgadas (se cumple con el requisito RNF-10: Requisito de diseño e implementación), sistemas operativos Android e iPhone y conexiones WiFi y de datos móviles de diversas compañías telefónicas (se cumple con el requisito RNF-06: Requisito de conexión).



Gráfica 6.3 - A3. Tipo de formación



Gráfica 6.4 - A4. Nivel de experiencia con smartphones



Gráfica 6.5 - A6. Conocimiento previo del robot

Por último en la pregunta A6 se comprueba si el usuario ya había trabajado previamente con el robot social empleado en el presente proyecto o si de lo contrario, no lo había visto nunca. Según la gráfica 6.5 la mayoría de los usuarios encuestados no habían visto nunca el robot.

En resumen, del perfil de usuario se puede concluir que cualquier usuario puede utilizar la aplicación y conseguir que el robot realice la tarea deseada, cumpliendo así con el requisito RNF-01 (Requisito de usabilidad), independientemente de la edad y formación de éste, así como del tipo de *smartphone* que utilice.

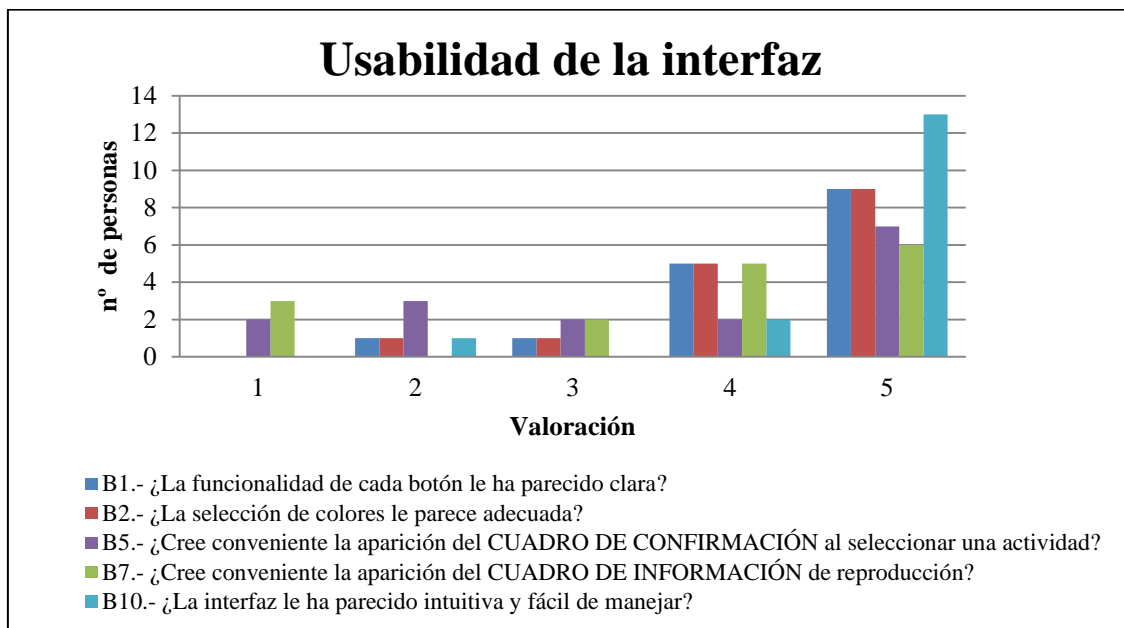
### 6.2.2 Usabilidad de la interfaz

En este apartado el usuario evalúa la usabilidad de la interfaz opinando acerca de los colores de la aplicación, la funcionalidad de los botones, la aparición de pop-ups de confirmación e información y del contenido del menú.

Además de las valoraciones recogidas en la gráfica 6.6 se han sugerido las siguientes ideas acerca de:

- Menú:
  - La mayoría de los usuarios consideran que el menú contiene todo lo necesario, pero hay quien piensa que sólo debería contener las funcionalidades del robot y quitar así la búsqueda de tareas, la selección de idioma y el acceso a la ayuda.
  - Agrupar las opciones del menú para que éste tuviera un tamaño más reducido.
  - Cuando el menú se encuentra desplegado, que se pliegue cuando se pulse en cualquier otro lugar de la pantalla y cuando se pulse atrás en el dispositivo.
  - Que se desplace la barra de búsqueda hacia arriba cuando aparece el teclado virtual del *smartphone* para que la barra no quede oculta detrás de éste (imagen de la derecha de la figura 5.31).
  - Indicaciones de *scroll* vertical ya que no se despliega completamente el menú y tanto el idioma como la ayuda quedan ocultas (imagen de la derecha de la figura 5.29).
  - Cambiar el color rojo.
- Cuadros de confirmación e información
  - Quitar el cuadro de confirmación e información (figura 5.34).
  - Sólo pedir confirmación para parar.
  - Sólo pedir confirmación cuando el robot está realizando otra tarea previamente.
  - Que el usuario pueda elegir si quiere que se muestre de nuevo el cuadro de información.
- Apariencia de la interfaz
  - La selección de colores de cada funcionalidad debería ser la misma.
  - Los botones de la funcionalidad multimedia se encuentran muy juntos (imagen de la derecha de la figura 5.30).

- Añadir texto a las imágenes de los botones “parar”, “pausar” y “reanudar” (imagen de la derecha de la figura 5.30).
- Añadir tamaños de letras y contrastes distintos.

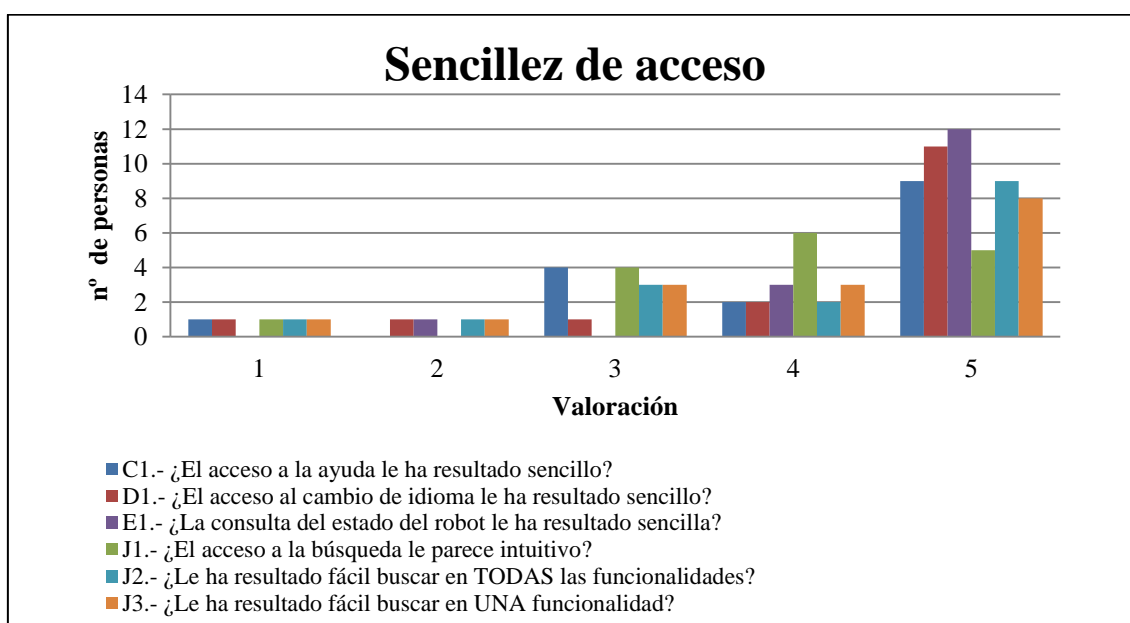


*Gráfica 6.6 - Usabilidad de la interfaz*

### 6.2.3 Funcionalidades

En este subapartado se evalúa que el uso de todas las funcionalidades de la aplicación sea sencillo e intuitivo.

En la gráfica 6.7 se valora la dificultad para acceder a la ayuda, al idioma, consultar el estado del robot y realizar una búsqueda.



*Gráfica 6.7 - Sencillez de acceso*

- **Búsqueda:**

Se observa que el acceso a la búsqueda no ha sido muy sencillo, debido a que la barra de búsqueda quedaba oculta detrás del teclado. Además no ha resultado fácil seguir el criterio de búsqueda: desde cualquier página que no fuera una de las funcionalidades del robot se realiza una búsqueda de todas las funcionalidades. Mientras que si el usuario se encuentra en una de las funcionalidades del robot, sólo se realiza la búsqueda para esa funcionalidad. Por esta razón, los resultados de búsqueda obtenidos no eran los que el usuario esperaba. Los usuarios han sugerido que se realice la búsqueda en todas las funcionalidades desde cualquier página de la aplicación.

- **Ayuda:**

A causa de que el menú no se desplegaba completamente, la ayuda quedaba oculta y a los usuarios les ha resultado difícil encontrarla.

Han sugerido:

- Poner la ayuda al pie de cada página en lugar de en el menú.
- Poner la ayuda dentro de cada funcionalidad.

- **Idioma:**

Al igual que la ayuda, quedaba oculto al desplegar el menú y a los usuarios les ha resultado difícil encontrarlo.

Han sugerido:

- Poner la selección del idioma al pie de cada página en lugar de en el menú.
- Poner además del texto del idioma, la bandera correspondiente para que se entienda de manera gráfica.

- **Consultar el estado:**

Acerca de la consulta del estado los usuarios creen más conveniente colocar el cuadro en el que se muestra el estado de la página principal, en la parte superior para que sea lo primero que se visualice.

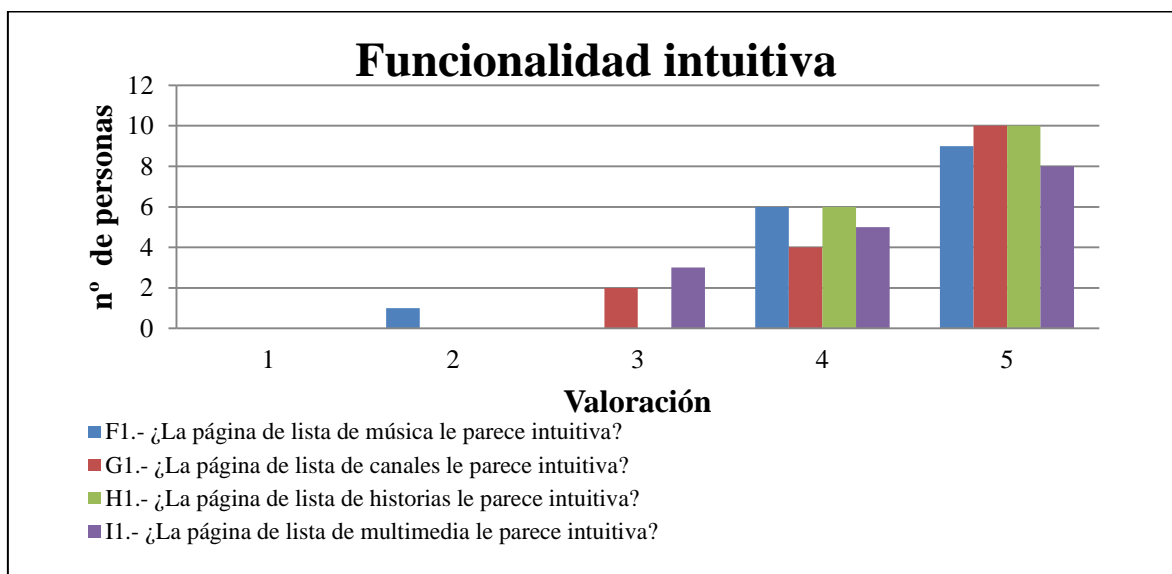
De la gráfica 6.8 se puede concluir que las páginas correspondientes a las funcionalidades música, fotos, tv/videos, noticias, historias y multimedia son muy intuitivas para el usuario.

Y de la gráfica 6.9 se puede confirmar que la realización de una tarea de cualquier funcionalidad es sencilla para el usuario.

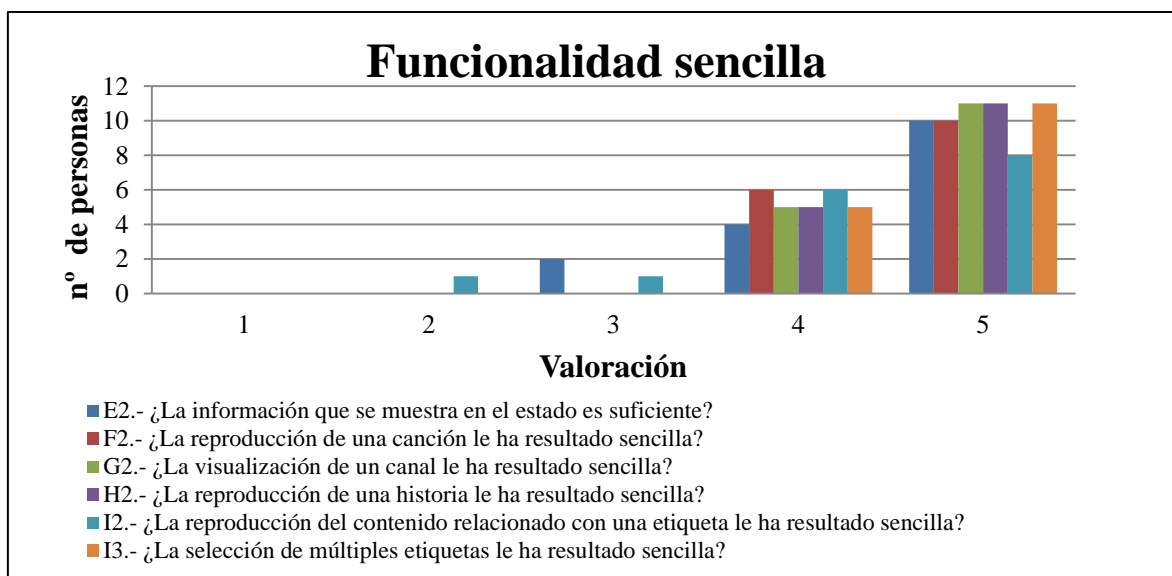
Se han propuesto las siguientes sugerencias para mejorar las funcionalidades:

- Lista de música: que sea posible pausar las canciones.
- Lista de canales: modo mosaico en el que se reproduzca unos segundos cada canal y se pase al siguiente.
- Lista de historias: poder elegir entre varias voces.
- Lista de contenidos multimedia: utilizar un único botón para aceptar la selección y reanudar la reproducción.





Gráfica 6.8 - Funcionalidad intuitiva



Gráfica 6.9 - Funcionalidad sencilla

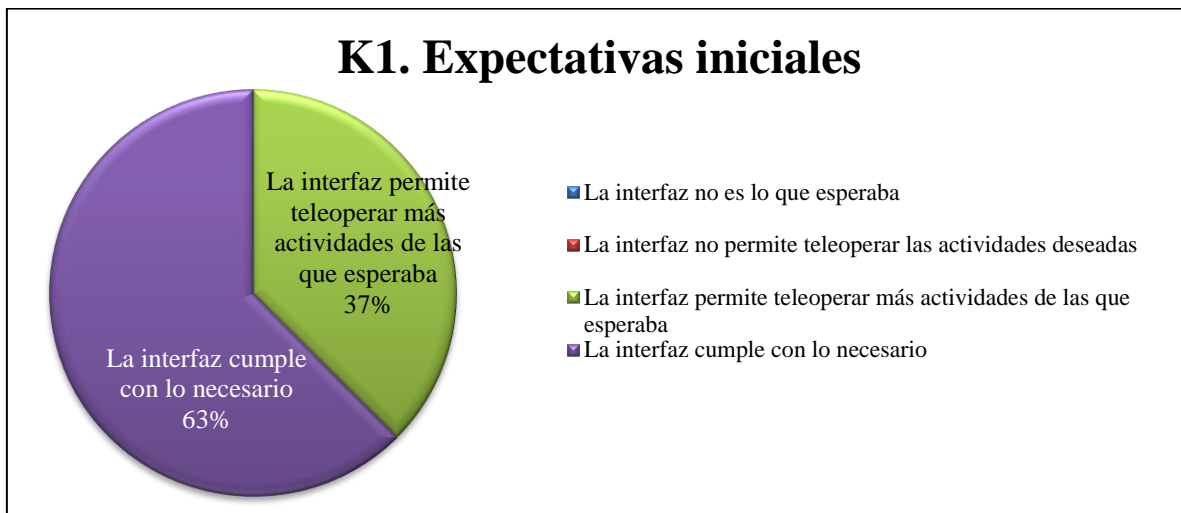
### 6.2.4 Valoración final

Como valoración final de la aplicación, el 37% de los usuarios encuestados ha superado las expectativas que tenían inicialmente acerca de la aplicación tal y como se muestra en la gráfica 6.10.

Con el fin de mejorar la interfaz se han dado las siguientes sugerencias:

- Indicar en el menú qué se está realizando.
- Añadir una barra de desplazamiento temporal para las funcionalidades de música y video.
- Mostrar las imágenes en la galería del dispositivo para la funcionalidad fotos.

- Identificar los botones “parar”, “pausar” y “reanudar” con texto.
- Añadir los botones de *play/stop* en la página principal para reanudar/parar las historias y los contenidos multimedia sin tener que acceder a sus páginas correspondientes.



*Gráfica 6.10 - K1. Expectativas iniciales*

Las mejoras que más han sugerido los usuarios tras valorar en conjunto los resultados de la encuesta realizada son las siguientes:

- Desplegar el menú completamente para que no queden ocultas las opciones de selección del idioma y la consulta de la ayuda.
- Cerrar el menú desplegado cuando se pulse fuera de éste o cuando se pulse atrás en el dispositivo móvil.
- Cambiar la barra de búsqueda para que no quede oculta detrás del teclado virtual del dispositivo.
- Cambiar el criterio de búsqueda para que busque en todas las funcionalidades independientemente de la página de la aplicación en la que se encuentre el usuario.
- Cambiar la posición del cuadro de información que indica el estado del robot a la parte superior de la página principal.
- Reducir el número de botones en la página de la funcionalidad multimedia.
- Añadir texto a la imagen de los botones “parar”, “pausar” y “reanudar”.
- Quitar el pop-up de información o ponerlo como opcional para el usuario.
- Quitar el pop-up de confirmación al seleccionar una tarea si el robot no está realizando ninguna tarea.

# Capítulo 7

## Mejoras y aplicaciones v.2.0 y v.2.1

En este capítulo se detallan las mejoras que incorporan las versiones 2.0 y 2.1 de la aplicación tras analizar las sugerencias de los diferentes usuarios encuestados con el fin de mejorar la versión 1.0. Además se explica el proceso de implementación de ambas versiones y finalmente, se realiza una comparativa de las tres.

## 7.1 Aplicación web (v2.0)

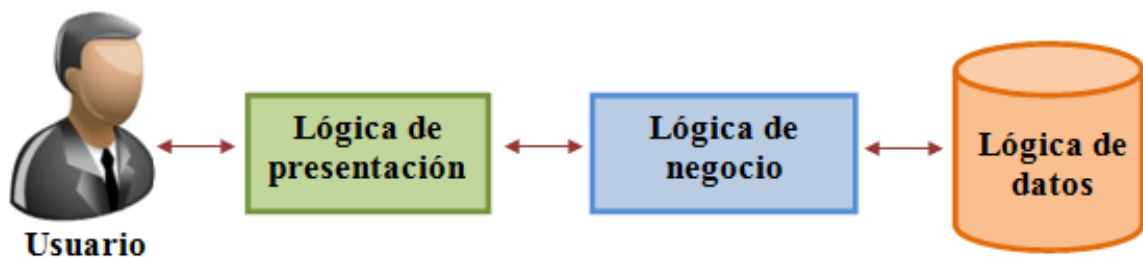
Tras finalizar el proceso de evaluación de la versión 1.0 de la aplicación con diferentes perfiles de usuario, se han analizado las sugerencias propuestas y detalladas en el apartado 6.2 de este proyecto para incorporarlas en la aplicación.

A pesar de que la primera versión cumple con todos los requisitos y casos de uso, se ha decidido realizar el rediseño de la aplicación, implementando una segunda versión, con el objetivo de mejorar la primera incorporando algunas sugerencias propuestas por los usuarios y permitiendo añadir más funcionalidades de los escenarios de entretenimiento y de otros, como por ejemplo el de seguridad, en trabajos futuros.

### 7.1.1 Diseño de la aplicación v2.0

En la segunda versión de la aplicación, se ha rediseñado únicamente la comunicación entre la aplicación y la base de datos (figura 5.5), dejando intacta la comunicación entre la aplicación y ROS. También se ha mantenido la funcionalidad de la primera versión, de manera que los cambios realizados son transparentes para el usuario.

Esta versión se ha realizado siguiendo el patrón de diseño dividido en las siguientes tres capas como se muestra en la figura 7.1:



*Figura 7.1 - Capas de la arquitectura cliente-servidor*

- **Lógica de presentación**

A esta capa también se la denomina capa de usuario debido a que se trata de la interfaz gráfica con la que el usuario va a interactuar. Por esta razón, debe ser entendible y fácil de usar para el usuario. Esta capa se ejecuta en el cliente y sólo se comunica con la lógica de negocio y las principales funciones que realiza son:

- Obtener información del usuario.
- Enviar la información recibida del usuario a la lógica de negocio para que ésta la procese.
- Recibir los resultados obtenidos del procesamiento de la información en la lógica de negocio.
- Presentar los resultados al usuario.

- **Lógica de negocio**

La lógica de negocio o capa de aplicación es la capa intermedia entre la lógica de presentación y la lógica de datos, es decir, actúa de puente entre el usuario y los datos. Esta capa se ejecuta tanto en el cliente como en el servidor. Sus principales tareas son:

- Recibir la información de usuario de la lógica de presentación.
- Interactuar con la lógica de datos para ejecutar las reglas de negocio que debe cumplir la aplicación.
- Enviar los resultados de procesamiento a la lógica de presentación.

- **Lógica de datos**

Esta capa se encarga de gestionar los datos a nivel de almacenamiento, es decir, se encarga de almacenar, recuperar, mantener y de asegurar la integridad de los datos. Puede estar formada por uno o más gestores de bases de datos. Se ejecuta en el servidor y se comunica únicamente con la lógica de negocio para recibir solicitudes de almacenamiento o recuperación de datos y devolver los resultados.

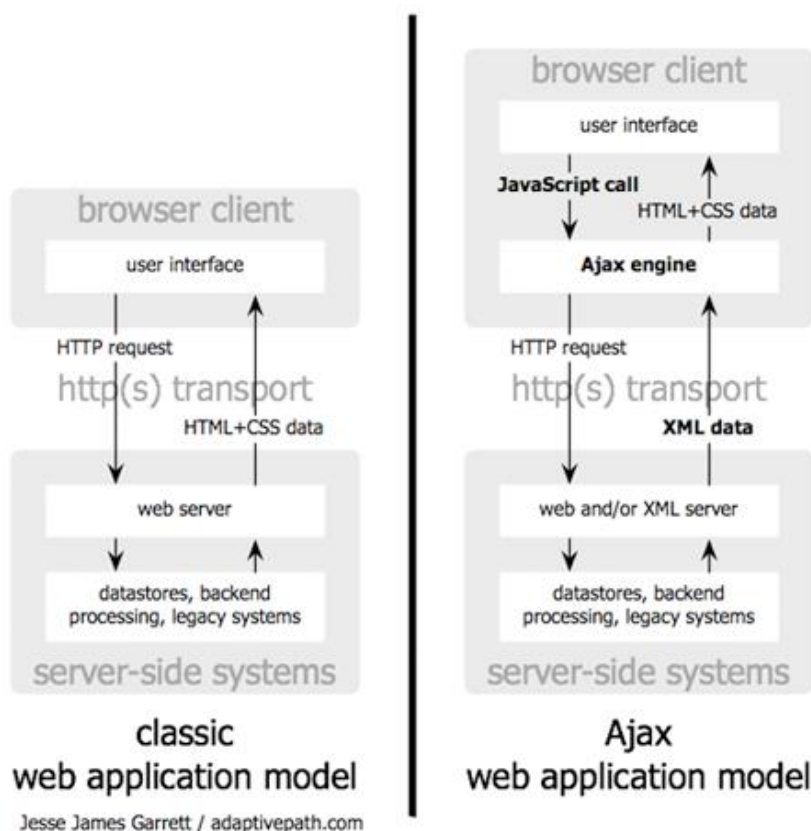
## 7.1.2 Implementación de la aplicación v.2.0

Para llevar a cabo la implementación de esta versión, se ha requerido una nueva tecnología, y una reestructuración de los ficheros de la aplicación que se detallan en los siguientes subapartados.

### 7.1.2.1 Tecnologías utilizadas

Además de las tecnologías de la primera versión, detalladas en el apartado 5.2, para esta versión se requiere la tecnología AJAX presentada por primera vez en el artículo “*Ajax: A New Approach to Web Applications*” [34], en el cuál AJAX se define como la unión de un conjunto de tecnologías independientes ya existentes:

- XHTML (eXtensible HyperText Markup Language) o HTML y CSS para desarrollar una presentación basada en estándares.
- DOM para visualizar e interactuar dinámicamente con la información presentada.
- XML (*eXtensible Markup Language* - Lenguaje de Marcas eXtensible), XSLT (*eXtensible Stylesheet Language Transformations*) y JSON para el intercambio y manipulación de datos.
- XMLHttpRequest para el intercambio de datos de manera asíncrona con el servidor web.
- JavaScript para unir las demás tecnologías.



*Figura 7.2 - Comparativa entre aplicación web clásica y aplicación web con AJAX*

En la figura 7.2 se puede observar la comparación entre el modelo de una aplicación web tradicional (aplicación v.1.0) y el modelo de una aplicación web con AJAX. La principal diferencia entre ambos modelos es la recarga de la página:

- En el modelo tradicional cuando se realiza una petición al servidor, el usuario debe esperar a que se recargue la página completamente, de forma que si se realizan peticiones de manera continua puede resultar algo molesto para el usuario. Para mejorar esta interacción entre el usuario y la aplicación se hace uso de una capa intermedia, AJAX, que intercambia la información con el servidor en segundo plano, eliminando la recarga constante de la página y mejorando la respuesta de la aplicación.
- En el modelo con AJAX se sustituyen las peticiones HTTP al servidor por peticiones JavaScript al elemento encargado de AJAX. Para peticiones simples no se requiere intervención del servidor por lo que la respuesta es inmediata. En caso de requerir una respuesta del servidor, se realiza una petición de manera asíncrona mediante AJAX sin interferir con la visualización ni el comportamiento de la página. Por tanto, utilizando este modelo, la interacción con el usuario no se ve afectada ni por las recargas de página ni por la espera de la respuesta del servidor.

En resumen, AJAX permite mejorar la interactividad, velocidad y usabilidad en las aplicaciones.

El lenguaje en el que se efectúan las funciones de llamada de AJAX es JavaScript, mientras que el acceso a los datos se realiza mediante el objeto XMLHttpRequest.

En la figura 7.3 se muestra como ejemplo el código de una función JavaScript que realiza una petición al servidor mediante AJAX. En el presente proyecto se hace uso de este tipo de funciones para cambiar el idioma de la aplicación y para mostrar las tablas que contienen la lista de tareas de cada funcionalidad.

```
function ajax () {  
  
    //Creación del objeto XMLHttpRequest en función del navegador utilizado  
    if (window.XMLHttpRequest) {  
        // code for IE7+, Firefox, Chrome, Opera, Safari  
        xmlhttp=new XMLHttpRequest();  
    } else {  
        // code for IE6, IE5  
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");  
    }  
  
    //Función a ejecutar cuando la petición haya finalizado y la respuesta  
    //está lista  
    xmlhttp.onreadystatechange=function() {  
        if (xmlhttp.status==200 && xmlhttp.readyState == 4){  
            document.getElementById("id").innerHTML=xmlhttp.responseText;  
        }  
    }  
  
    //Envío de la petición al servidor de forma asíncrona  
    xmlhttp.open("GET","pagina.php",true);  
    xmlhttp.send();  
}
```

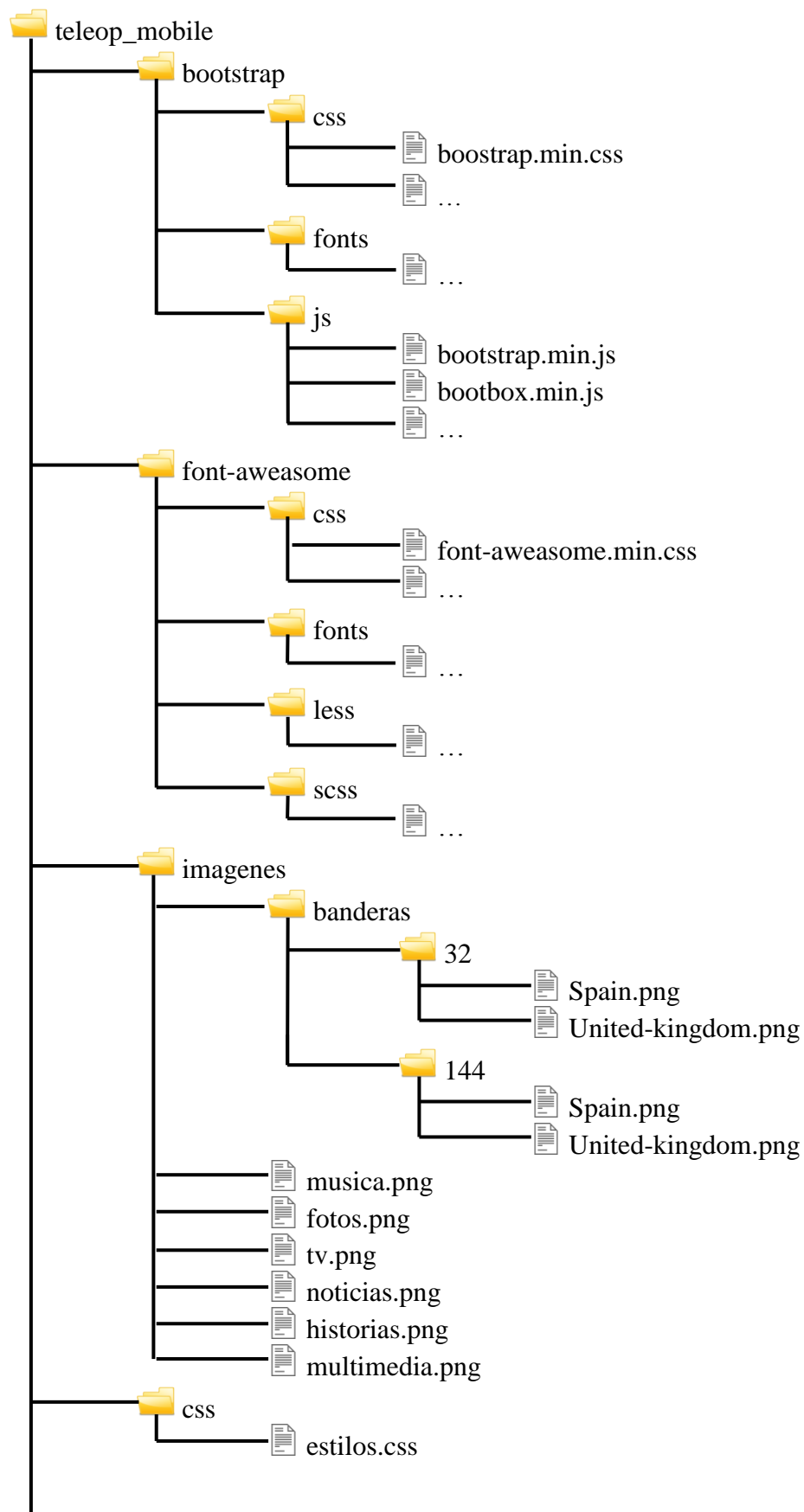
*Figura 7.3 - Código de una función AJAX de ejemplo*

### 7.1.2.2 Estructura de ficheros

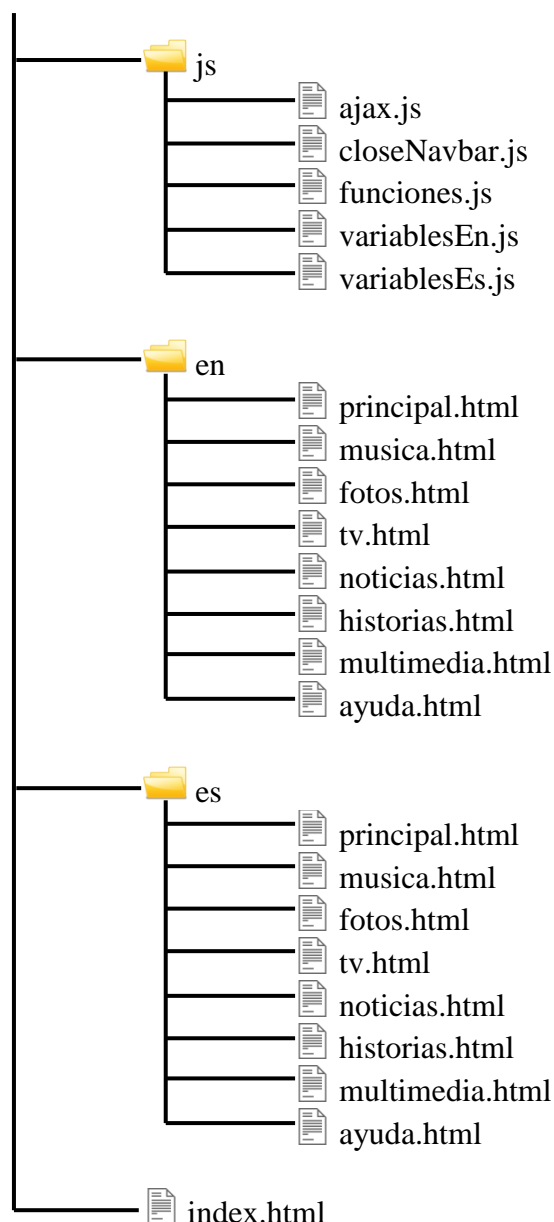
La estructura de ficheros de esta versión es un factor clave para llevar a cabo una implementación sencilla, fácil de mantener y de ampliar. Se ha separado por un lado la parte del cliente que estará formada por ficheros HTML, JavaScript y CSS y por otro lado la parte del servidor que está formada por ficheros PHP que se encargan de acceder a la base de datos MySQL.

- **Ficheros en el cliente**

La estructura de directorios representada en la figura 7.4, puede estar almacenada tanto en el servidor web como en un equipo remoto, ya que accede al contenido del servidor por medio de AJAX.







*Figura 7.4 - Estructura de ficheros del cliente*

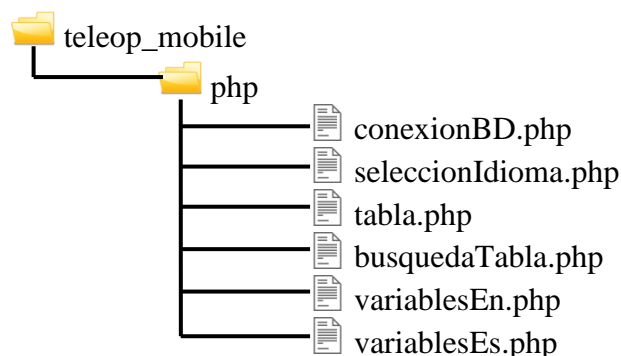
Respecto a la estructura de ficheros de la aplicación v.1.0, se incorporan en la carpeta “imagenes”, las imágenes correspondientes a las banderas en dos tamaños, uno para la página de inicio y el otro para la selección de idioma del menú desplegable. Se crea un directorio por cada idioma disponible en la aplicación (en este caso, “es” para español y “en” para inglés) que contienen las páginas de la aplicación.

Y en la carpeta “js” se añade:

- El fichero ajax.js con las funciones AJAX cuyo código es similar al del ejemplo de la figura 7.3.
- El fichero closeNavbar.js para plegar el menú cuando se pulse en otro lugar de la pantalla.
- Los ficheros variablesEn.js y variablesEs.js que contienen las variables correspondientes al idioma seleccionado.

### ▪ Ficheros en el servidor

La estructura de directorios en el lado del servidor (figura 7.5) debe estar almacenada en la carpeta raíz del servidor web. En el caso de Apache2.2 la carpeta raíz por defecto es /var/www, por lo que en dicho directorio es donde deberían guardarse los siguientes directorios y ficheros:



*Figura 7.5 - Estructura de ficheros del servidor*

En esta versión únicamente hay seis ficheros con código PHP que deben estar en el servidor.

- conexionBD.php: realiza la conexión con la base de datos *robalz* del robot *Mini*.
- seleccionIdioma.php: carga las variables correspondientes al idioma seleccionado por el usuario para cambiar el idioma del texto que aparece en la aplicación.
- tabla.php: consulta la base de datos para mostrar las tablas con las tareas disponibles de cada funcionalidad.
- busquedaTabla.php: consulta la base de datos para mostrar las tareas que coinciden con el texto introducido por el usuario.

### 7.1.3 Mejoras en v.2.0

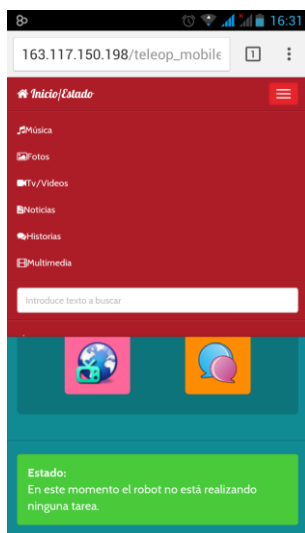
Respecto a la primera, la segunda versión de la aplicación incorpora las siguientes mejoras:

- Se ha añadido una página inicial al acceder a la aplicación para que el usuario elija el idioma deseado al entrar en la aplicación (figura 7.6).

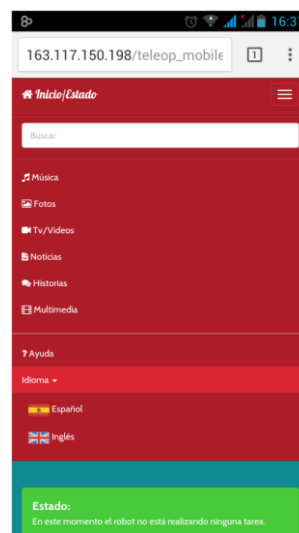


*Figura 7.6 - Mejoras v.2.0: Página inicial*

#### APLICACIÓN WEB V.1.0

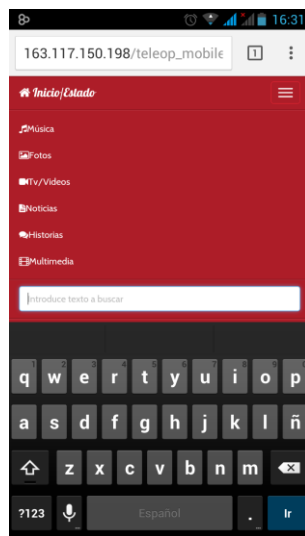
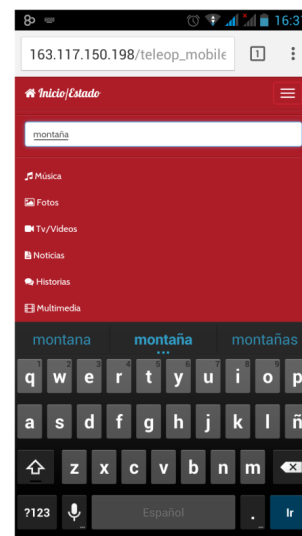
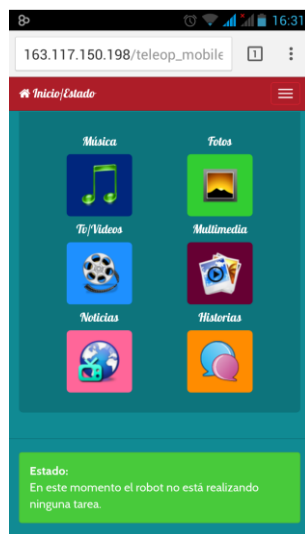
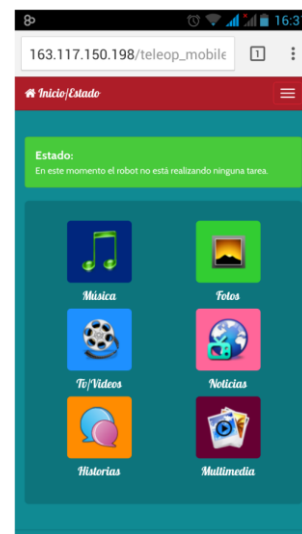


#### APLICACIÓN WEB V.2.0



*Figura 7.7 - Mejoras v.2.0: Menú desplegado*

- Se han añadido banderas en la selección del idioma para que se entienda de forma gráfica (figura 7.7).
- Se ha simplificado el texto de ayuda.
- El menú desplegado se cierra cuando el usuario pulsa en otro lugar de la pantalla o pulsa “atrás” en su dispositivo móvil.
- El menú se despliega completamente de forma que las opciones “Idioma” y “Ayuda” ya no quedan ocultas. Además si se realiza *scroll* vertical, al final del menú desplegado aparece la página actual en la que se encuentra el usuario (figura 7.7).
- La barra de búsqueda se ha colocado en la parte superior del menú desplegable para que no se oculte tras el teclado virtual (figura 7.8).
- El cuadro de estado de la página principal se ha colocado en la parte superior (figura 7.9).

**APLICACIÓN WEB V.1.0****APLICACIÓN WEB V.2.0***Figura 7.8 - Mejoras v.2.0: Barra de búsqueda***APLICACIÓN WEB V.1.0****APLICACIÓN WEB V.2.0***Figura 7.9 - Mejoras v.2.0: Estado del robot*

- Se ha reducido el uso de botones en la funcionalidad multimedia (figura 7.10).
- Se ha añadido un cuadro de información en la funcionalidad multimedia que aparece cuando el usuario pulsa “Aceptar selección” sin seleccionar ninguna etiqueta (figura 7.11).

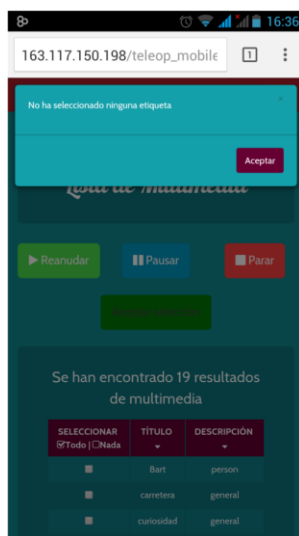
## APLICACIÓN WEB V.1.0



## APLICACIÓN WEB V.2.0



*Figura 7.10 - Mejoras v.2.0: Botones funcionalidad multimedia*



*Figura 7.11 - Mejoras v.2.0: Cuadro información en multimedia*

- Se ha añadido texto a los botones “reanudar”, “pausar” y “parar” de cada una de las funcionalidades (figura 7.12).
- Las tablas, por defecto, se ordenan alfabéticamente (A-Z) por título (ejemplo: figura 7.12).
- Se ha eliminado el criterio de búsqueda separada. En esta versión cuando se realiza una búsqueda, se busca en todas las funcionalidades independientemente de la página en la que se encuentre el usuario y no distingue entre mayúsculas, minúsculas ni tildes.

## APLICACIÓN WEB V.1.0



## APLICACIÓN WEB V.2.0



*Figura 7.12 - Mejoras v.2.0: Identificación de botones*

## 7.2 Aplicación híbrida (v2.1)

En la versión 2.0 se ha separado la parte del cliente de la del servidor de manera que la aplicación se puede ejecutar desde un equipo remoto al servidor. Gracias a esta modificación ha sido posible adaptar la aplicación web en aplicación híbrida utilizando Phonegap/Cordova.

### 7.2.1 Implementación de la aplicación v.2.1

Para crear la aplicación híbrida se debe seguir la guía de documentación de Phonegap/Cordova [35]. A continuación se explican los principales pasos a realizar para crear la aplicación híbrida.

En primer lugar se debe instalar Android Studio de [36], instalar Eclipse de [37], añadir el plugin ADT [38] y configurar las variables de entorno de *java* y *ant*.

Una vez se han cumplido con los requisitos, se crea el proyecto. Primero se debe ejecutar un terminal y acceder al directorio en el que se desea crear el proyecto. A continuación se deben ejecutar los siguientes comandos para crear el proyecto “teleopMobile” y añadir la plataforma Android:

```
$ cordova create teleopMobile com.teleop.mobile TeleopMobile
$ cd teleopMobile
$ cordova platform add android
$ cordova build
```

Después de crear el proyecto se ejecuta Eclipse y se siguen los siguientes pasos:

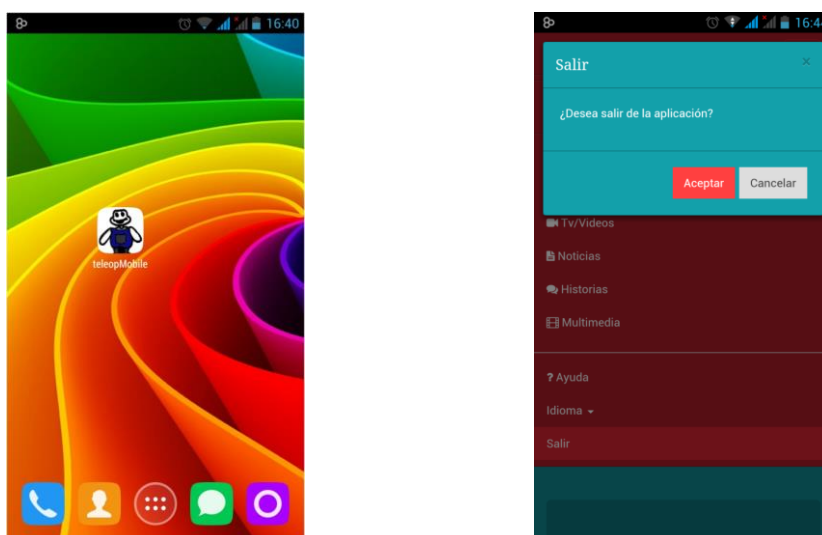
- New → Project → Android Project from Existing Code → elegir el directorio en el que se ha creado el proyecto de Cordova.
- Botón derecho sobre el proyecto → Properties → Resource → Resource Filters → eliminar todo
- Botón derecho sobre el proyecto → Properties → Android → escoger la API Android deseada.

Tras realizar todos los pasos del procedimiento descrito, se añade el código de la aplicación, se realizan los cambios oportunos y se ejecuta para generar la aplicación con extensión .apk, la cual será instalada en el dispositivo móvil Android.

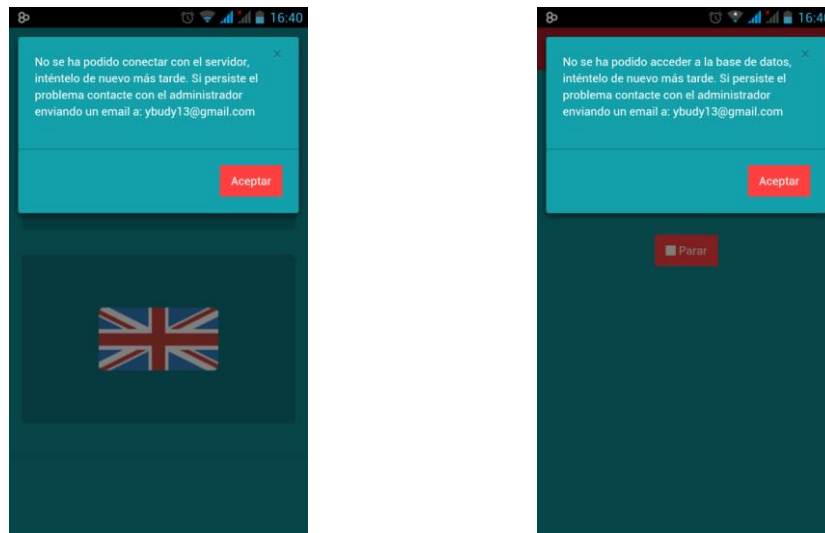
## 7.2.2 Mejoras en v.2.1

Esta versión incorpora las mejoras de la versión 2.0 y añade las siguientes:

- Acceso a la aplicación desde un icono en el dispositivo, sin necesidad de introducir la URL en el navegador web (imagen izquierda de la figura 7.13).
- Se añade la opción de salir de la aplicación en el menú de la aplicación. (imagen derecha de la figura 7.13).
- Se añaden mensajes de error de conexión con el servidor y con la base de datos para informar al usuario. (figura 7.14).



*Figura 7.13 - Mejoras v.2.1: Acceso a la aplicación (izq.) y Salir (dcha.)*



*Figura 7.14 - Mejoras v.2.1: Mensaje error de conexión con el servidor (izq.) y con la base de datos (dcha.)*

## 7.3 Comparativa de versiones

En este apartado se realiza un breve resumen de las diferencias en la implementación de cada versión de la aplicación.

### 7.3.1 Versión 1.0

En la primera versión se desarrolla el modelo tradicional en los lenguajes HTML, CSS, JavaScript y PHP. Se trata de una aplicación del lado del servidor y requiere la carga completa de la página cuando se solicita un recurso del servidor.

### 7.3.2 Versión 2.0

Con el objetivo de mejorar la interacción entre el usuario y la interfaz se implementa una segunda versión de la aplicación utilizando el modelo con AJAX descrito en el subapartado 7.1.2.1 de este capítulo, de manera que se elimina la recarga completa de la página web cuando se pide un recurso al servidor. Se separa la parte del cliente de la del servidor siguiendo la estructura de ficheros detallada en la figura 7.4 y figura 7.5, por lo que la parte del cliente desarrollada en HTML, CSS y JavaScript puede estar en un equipo remoto al servidor y accederá a éste para solicitar las páginas con código PHP necesarias.

### 7.3.3 Versión 2.1

Mediante la versión 2.0 se ha realizado la adaptación a aplicación híbrida para el SO Android utilizando Phonegap/Cordova. Para ello se ha requerido el entorno de desarrollo Eclipse junto con el SDK y ADT de Android.



### 7.3.4 Comparación

Todas las versiones tienen la misma funcionalidad y cumplen los requisitos y casos de uso definidos, de manera que los cambios son transparentes para el usuario, salvo las mejoras sugeridas tras la encuesta realizada. También se ha mantenido intacta la parte de comunicación entre la aplicación y ROS.

Se han realizado las pruebas unitarias descritas en el apartado 6.1 de forma paralela a la implementación de todas las versiones, aunque sólo la primera versión ha sido probada con un conjunto de usuarios ajenos al proyecto.

Cabe destacar que utilizando el *framework* Bootstrap, la aplicación se ha adaptado a diferentes tamaños de pantalla en todas las versiones.

En la tabla 7.1 se puede observar el resumen de las características que diferencian a las tres versiones implementadas.

	1ª versión (1.0)	2ª versión (2.0)	3ª versión (2.1)
<b>Sistema Operativo</b>	Todos	Todos	Android
<b>Tipo de aplicación</b>	Web	Web	Híbrida
<b>Lugar aplicación</b>	Lado del servidor	Lado del cliente o servidor	Lado del cliente
<b>Modelo</b>	Tradicional	Con AJAX	Con AJAX
<b>Uso navegador web</b>	Sí	Sí	No
<b>Instalación</b>	No	No	Sí
<b>Acceso al hardware del dispositivo</b>	No	No	Sí
<b>Mantenibilidad</b>	Inmediata	Inmediata	Reinstalar la aplicación (Actualizar)

*Tabla 7.1 - Comparativa de las características de las versiones implementadas*

En el anexo B de este documento, se detalla el manual de usuario con los pasos a seguir para utilizar las aplicaciones v.2.0 y v.2.1.



# Capítulo 8

## Gestión del proyecto

En este capítulo se detallan las tareas en las que han sido divididas cada una de las fases que componen el proyecto, así como su planificación temporal inicial y final y el presupuesto necesario para llevarlo a cabo.

## 8.1 Fases del desarrollo

El proyecto se ha dividido en diversas fases, siguiendo un modelo de desarrollo evolutivo como se muestra en la figura 8.1: planificación, análisis de requisitos, diseño, desarrollo, integración y pruebas, implementación y mantenimiento.



*Figura 8.1 - Esquema del ciclo de vida evolutivo*

El modelo de desarrollo evolutivo o desarrollo con prototipación, se inicia con los objetivos generales del software, para posteriormente identificar los requisitos. Se realiza un prototipo del software para que el usuario lo evalúe y cuando éste esté satisfecho, se lleve a cabo la implementación del software final.

## 8.2 Diagramas de Gantt

Una vez definido el modelo del proyecto se realiza la planificación del mismo. Para ello se utiliza el diagrama de Gantt que permite mostrar el tiempo previsto para la realización de cada tarea.

En los subapartados 8.2.1 y 8.2.2 se muestran los diagramas de Gantt de la planificación inicial y final del proyecto respectivamente. Para la realización de ambos diagramas se ha utilizado el software de código abierto *Gantt Project*.

### 8.2.1 Planificación inicial

En este subapartado se muestra el diagrama de Gantt inicial que muestra las tareas de cada fase planificadas previamente al desarrollo del proyecto.

El calendario laboral comprende los siete días de la semana y cada día de trabajo corresponde a tres horas, excepto la documentación de tareas que corresponde a media hora de trabajo.

En la figura 8.2 se pueden observar cinco columnas: nombre de la tarea, cargo de la persona que realiza la tarea, fecha de inicio, fecha de fin y duración de la tarea. Y en la figura 8.3 se muestra el diagrama de Gantt completo.

La finalización del proyecto estaba prevista para el mes de octubre de 2014, con un periodo de tiempo intermedio de pausa para la preparación de los exámenes de las asignaturas de la carrera y por cierre de la universidad.



Nombre	Cargo	Fecha de inicio	Fecha de fin	Duración
♀ • Inicio del proyecto		30/01/2014	02/02/2014	
• Instalación del entorno	Programador	30/01/2014	02/02/2014	4
♀ • Análisis		03/02/2014	15/02/2014	
• Estudio de ROS	Analista	03/02/2014	05/02/2014	3
• Definición de las funcionalidades	Analista	06/02/2014	07/02/2014	2
• Descripción de casos de uso	Analista	08/02/2014	11/02/2014	4
• Descripción de requisitos	Analista	12/02/2014	15/02/2014	4
♀ • Diseño		16/02/2014	26/02/2014	
• Diseño de las páginas de la interfaz	Investigador	16/02/2014	20/02/2014	5
• Diseño de esquema y funcionalidad de la ...	Investigador	21/02/2014	26/02/2014	6
♀ • Implementación		27/02/2014	18/09/2014	
• Comparativa de aplicaciones móviles	Analista	27/02/2014	03/03/2014	5
• Comparativa Rosbridge y Rosjava	Analista	04/03/2014	06/03/2014	3
• Elección del tipo de aplicación móvil	Investigador	13/03/2014	15/03/2014	3
• Comparativa de Frameworks	Analista	07/03/2014	12/03/2014	6
• Elección del Framework	Investigador	16/03/2014	19/03/2014	4
• Implementación de las páginas de la aplic...	Programador	20/03/2014	21/04/2014	33
• Implementación de mejoras	Programador	11/09/2014	18/09/2014	8
♀ • Pruebas		27/03/2014	25/09/2014	
• Pruebas unitarias	Programador	27/03/2014	21/04/2014	26
• Pruebas finales de la aplicación v1.0	Programador	24/04/2014	08/05/2014	14
• Evaluación con usuarios	Programador	01/09/2014	07/09/2014	7
• Análisis de resultados	Analista	08/09/2014	10/09/2014	3
• Pruebas finales de la aplicación final	Programador	19/09/2014	25/09/2014	7
♀ • Documentación		03/02/2014	26/10/2014	
• Documentar tareas	Documentador	03/02/2014	08/05/2014	94
• Documentar más tareas	Documentador	01/09/2014	19/09/2014	19
• Memoria	Documentador	10/07/2014	30/09/2014	82
• Presentación	Documentador	20/10/2014	26/10/2014	7

*Figura 8.2 - Diagrama de Gantt: tareas de la planificación inicial*

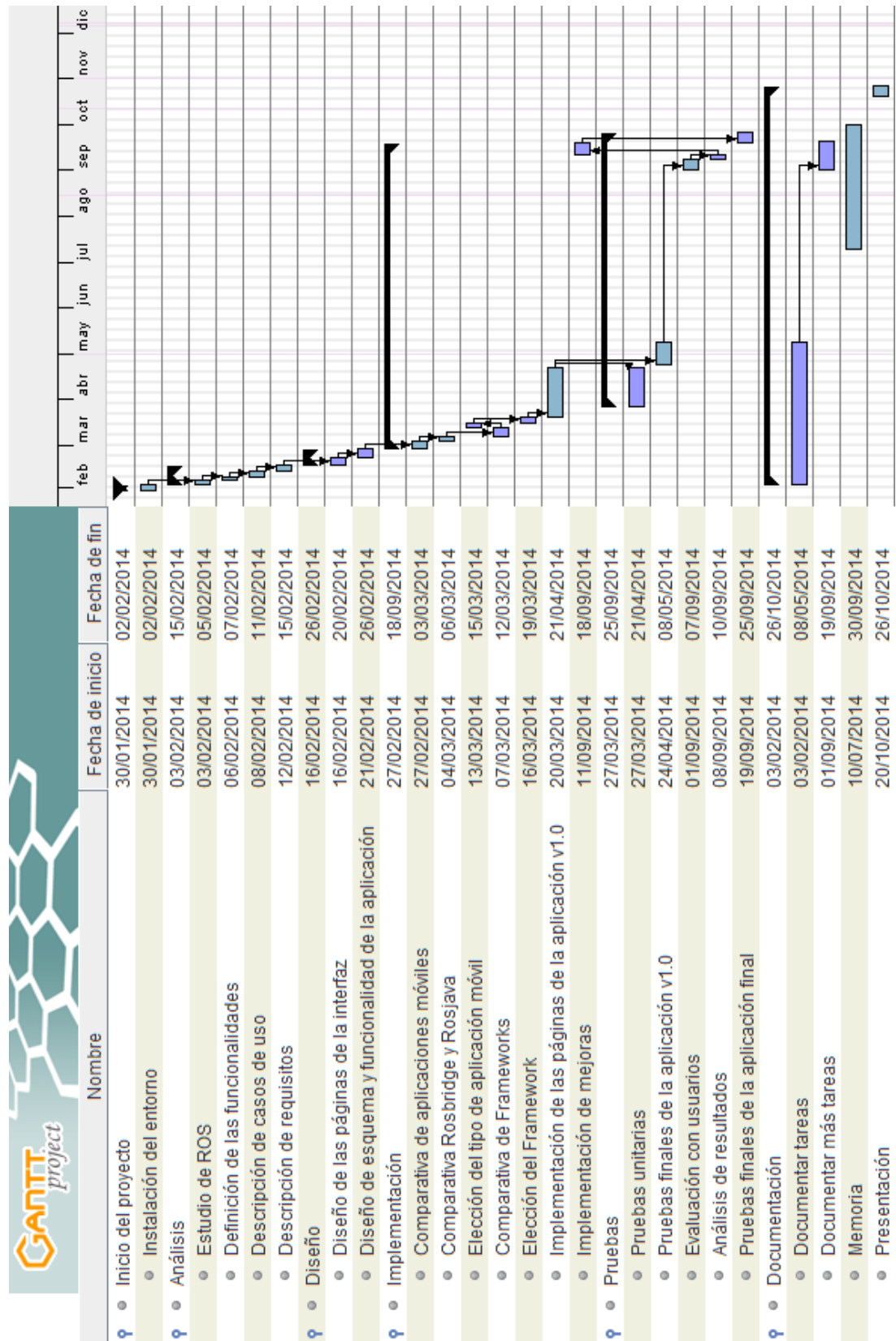


Figura 8.3 - Diagrama de Gantt de la planificación inicial

## 8.2.2 Planificación final

En este subapartado se muestra el diagrama de Gantt final y por tanto el que muestra la planificación real de cada una de las tareas.

El calendario laboral comprende los siete días de la semana y cada día de trabajo corresponde a tres horas, excepto la documentación de tareas que corresponde a media hora de trabajo.

En la figura 8.4 se pueden observar cinco columnas: nombre de la tarea, cargo de la persona que realiza la tarea, fecha de inicio, fecha de fin y duración de la tarea.



	Nombre	Cargo	Fecha de inicio	Fecha de fin	Duración
♀	• Inicio del proyecto		30/01/2014	05/02/2014	
	• Instalación del entorno	Programador	30/01/2014	05/02/2014	7
♀	• Análisis		06/02/2014	21/02/2014	
	• Estudio de ROS	Analista	06/02/2014	12/02/2014	7
	• Definición de las funcionalidades	Analista	13/02/2014	13/02/2014	1
	• Descripción de casos de uso	Analista	14/02/2014	17/02/2014	4
	• Descripción de requisitos	Analista	18/02/2014	21/02/2014	4
♀	• Diseño		22/02/2014	28/02/2014	
	• Diseño de las páginas de la interfaz	Investigador	22/02/2014	24/02/2014	3
	• Diseño de esquema y funcionalidad de la aplicación	Investigador	25/02/2014	28/02/2014	4
♀	• Implementación		01/03/2014	27/02/2015	
	• Comparativa de aplicaciones móviles	Analista	01/03/2014	03/03/2014	3
	• Comparativa Rosbridge y Rosjava	Analista	04/03/2014	05/03/2014	2
	• Elección del tipo de aplicación móvil	Investigador	06/03/2014	08/03/2014	3
	• Comparativa de Frameworks	Analista	09/03/2014	13/03/2014	5
	• Elección del Framework	Investigador	14/03/2014	17/03/2014	4
	• Implementación de las páginas de la aplicación v1.0	Programador	18/03/2014	23/04/2014	37
	• Implementación de más funcionalidades	Programador	08/09/2014	11/10/2014	34
	• Diseño y creación de las tablas y vistas MySQL	Operador	13/10/2014	20/10/2014	8
	• Diseño y creación de los servicios y temas de ROS	Programador	21/10/2014	30/11/2014	40
	• Modificación del modelo de la aplicación v.1.0	Investigador	19/01/2015	30/01/2015	12
	• Implementación de la aplicación v2.0	Programador	02/02/2015	20/02/2015	19
	• Paso de la aplicación a híbrida (v2.1)	Programador	21/02/2015	27/02/2015	7
♀	• Pruebas		25/03/2014	14/03/2015	
	• Pruebas unitarias de las páginas de la aplicación v.1.0	Programador	25/03/2014	23/04/2014	30
	• Pruebas unitarias de más funcionalidades	Programador	15/09/2014	30/11/2014	75
	• Pruebas finales de la aplicación v1.0	Programador	01/12/2014	06/12/2014	5
	• Evaluación con usuarios	Analista	14/12/2014	19/12/2014	6
	• Análisis de resultados	Analista	20/12/2014	26/12/2014	6
	• Pruebas finales de la aplicación v2.0	Programador	03/03/2015	14/03/2015	12
	• Pruebas finales de la aplicación v2.1	Programador	03/03/2015	14/03/2015	12
♀	• Documentación		05/02/2014	15/06/2015	
	• Documentar tareas	Documentador	05/02/2014	23/04/2014	78
	• Documentar más tareas	Documentador	08/09/2014	26/02/2015	165
	• Memoria	Documentador	27/02/2015	07/06/2015	100
	• Presentación	Documentador	07/06/2015	15/06/2015	9

Figura 8.4 - Diagrama de Gantt: tareas planificación final

El proyecto se divide en seis etapas, que se dividen a su vez en varias tareas:

- Inicio del proyecto:
  - Instalación del entorno: obtener e instalar el software necesario para poder llevar a cabo la implementación de la aplicación.
- Análisis:
  - Estudio de ROS: estudiar y familiarizarse con las librerías y herramientas del *framework* ROS.
  - Definición de las funcionalidades: estudiar el funcionamiento, características físicas, comportamientos básicos y funcionalidades existentes del robot social.
  - Descripción de casos de uso: definir los casos de uso en función de las funcionalidades que debe implementar la aplicación móvil.
  - Descripción de requisitos: analizar los requisitos que debe cumplir la aplicación móvil.
- Diseño
  - Diseño de las páginas de la interfaz: diseñar el contenido y formato de cada una de las páginas que componen la interfaz.
  - Diseño de esquema y funcionalidad de la aplicación: teniendo en cuenta los requisitos definidos en la fase anterior, diseñar el esquema principal del sistema y la funcionalidad de la aplicación siguiendo los casos de uso.
- Implementación
  - Comparativa de aplicaciones móviles: estudiar los tipos de aplicaciones para dispositivos móviles.
  - Comparativa *rosbridge* y *rosjava*: estudiar la comunicación entre las aplicaciones móviles y ROS.
  - Elección del tipo de aplicación móvil: escoger la aplicación móvil más adecuada teniendo en cuenta la comparativa de las aplicaciones.
  - Comparativa de *frameworks*: estudiar los diferentes *frameworks* disponibles que mejor se adapten a las funcionalidades de la aplicación móvil.
  - Elección del *framework*: elegir el *framework* que permita realizar una interfaz atractiva, sencilla y agradable para el usuario y pueda incorporar todas las funcionalidades requeridas.
  - Implementación de las páginas de la aplicación v1.0: codificar la primera versión de la aplicación.
  - Implementación de más funcionalidades: incorporar más funcionalidades con la finalidad de mejorar la usabilidad.



- Diseño y creación de tablas y vistas MySQL: estudiar y familiarizarse con la base de datos, para acceder y crear las vistas y tablas necesarias para mostrar y almacenar la información relevante de la aplicación.
  - Diseño y creación de los servicios y temas ROS: diseñar y codificar los servicios y temas de ROS necesarios para la comunicación entre la aplicación y el robot.
  - Modificación del modelo de la aplicación: rediseñar la aplicación v1.0 con modelo tradicional para utilizar el modelo con AJAX.
  - Implementación de la aplicación v2.0: incorporar las sugerencias de los usuarios a la aplicación utilizando el modelo con AJAX.
  - Paso de la aplicación a híbrida v2.1: convertir la aplicación web en una aplicación híbrida para Android.
- Pruebas
    - Pruebas unitarias: probar cada funcionalidad por separado para verificar que funcionan correctamente individualmente.
    - Pruebas unitarias de más funcionalidades: continuar con las pruebas de cada funcionalidad.
    - Pruebas finales v1.0: probar la aplicación completa para verificar el correcto funcionamiento del sistema.
    - Evaluación con usuarios: probar la aplicación v1.0 con diversos usuarios para evaluar el funcionamiento final del sistema.
    - Análisis de resultados: analizar los resultados para incorporar las posibles mejoras a la aplicación.
    - Pruebas finales v2.0: probar la aplicación con modelo AJAX.
    - Pruebas finales v2.1: probar la aplicación híbrida.
  - Documentación: documentar los aspectos más relevantes para comprender el desarrollo y funcionamiento del proyecto.

Comparando el diagrama inicial y el final, la finalización del proyecto se ha retrasado ocho meses debido a la incorporación de algunas mejoras sugeridas por los usuarios encuestados y a la realización de nuevas tareas que no estaban previstas en la planificación inicial, como la creación de vistas y tablas MySQL, la creación de servicios y temas ROS y el rediseño del modelo de la aplicación para mejorar la interacción entre el usuario y la aplicación. Además, hay periodos temporales en los cuales se ha parado el proyecto debido a la necesidad de compaginar el desarrollo del proyecto con los estudios de las últimas asignaturas de la carrera.

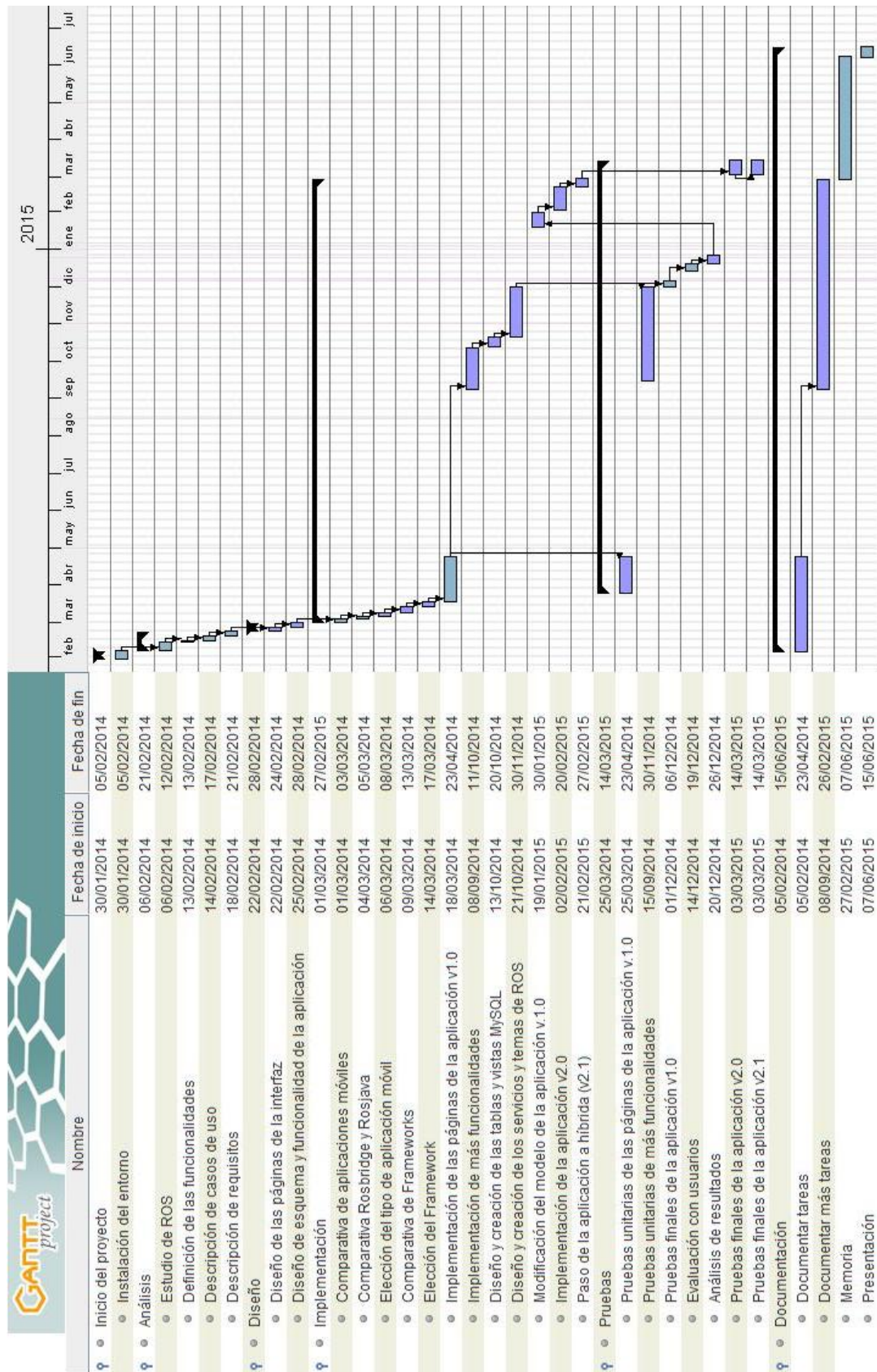


Figura 8.5 - Diagrama de Gantt de la planificación final

## 8.3 Presupuesto del proyecto

En este apartado se muestra el desglose de los costes asociados necesarios para la realización del proyecto tanto para la planificación inicial como para la final.

### 8.3.1 Horas dedicadas

En base a los diagramas de Gantt de la figura 8.3 y figura 8.5, se han calculado el número de horas totales dedicadas en cada fase del proyecto de cada planificación. En la tabla 8.1 se muestran los días y horas correspondientes a cada fase de la planificación inicial y en la tabla 8.2 se muestran los de la planificación final.

Fase		Nº de días	Nº de horas/día	Nº de horas
Inicio		4	3	12
Análisis		13	3	39
Diseño		11	3	33
Implementación		62	3	186
Pruebas		57	3	171
Documentación	Tareas	113	0.5	56.5
	Memoria	89	3	267
<b>TOTAL</b>				<b>764.5 horas</b>

*Tabla 8.1 - Planificación inicial: Resumen de horas dedicadas*

Fase		Nº de días	Nº de horas/día	Nº de horas
Inicio		7	3	21
Análisis		16	3	48
Diseño		7	3	21
Implementación		174	3	522
Pruebas		146	3	438
Documentación	Tareas	243	0.5	121.5
	Memoria	109	3	327
<b>TOTAL</b>				<b>1498.5 horas</b>

*Tabla 8.2 - Planificación final: Resumen de horas dedicadas*

Por tanto, el tiempo dedicado al proyecto es la suma del tiempo dedicado a cada una de las fases en las que se ha dividido. El coste inicial en horas de la totalidad del proyecto es de 764,5 y el coste final es de 1498,5.

### 8.3.2 Personal

En la tabla 8.3 y tabla 8.4 se detalla el salario de los diferentes cargos correspondientes al personal encargado de realizar las diversas tareas del proyecto para ambas planificaciones. Los salarios mensuales de cada cargo se han consultado las tablas salariales para los años 2014 y 2015 en la sección III del BOE (Boletín Oficial del Estado) de febrero de 2015 (referencias [39] y [40]).

Cargo	Salario 2014 (€/mes)	Nº meses en 2014	Salario mensual (€/mes)
Jefe de proyecto	1417,47	5	1417,47
Analista	1539,14	5	1539,14
Investigador	1584,75	5	1584,75
Programador	1180,84	5	1180,84
Documentador	1530	5	1530,00

*Tabla 8.3 - Planificación inicial: Salario mensual de cada cargo*

Cargo	Salario 2014 (€/mes)	Salario 2015 (€/mes)	Nº meses en 2014	Nº meses en 2015	Salario mensual (€/mes)
Jefe de proyecto	1417,47	1425,97	7	6	1421,39
Analista	1539,14	1548,37	7	6	1543,40
Investigador	1584,75	1594,26	7	6	1589,14
Programador	1180,84	1180,84	7	6	1180,84
Operador	821,69	821,69	7	6	821,69
Documentador	1530	1530	7	6	1530,00

*Tabla 8.4 - Planificación final: Salario mensual de cada cargo*

Para obtener el salario mensual de cada cargo se debe tener en cuenta el número de meses trabajados en cada año y el salario mensual de ambos. La ecuación 8.1 muestra la forma de obtener el salario mensual.

$$\text{Salario mensual} = \frac{(\text{salario 2014})(n^{\circ} \text{ meses 2014}) + (\text{salario 2015})(n^{\circ} \text{ meses 2015})}{n^{\circ} \text{ total de meses}}$$

*Ecuación 8.1 - Cálculo del salario mensual*

Tras conocer cada uno de los salarios mensuales, se realiza el cálculo del coste total del personal en función de la dedicación de cada cargo en el proyecto. En la tabla 8.5 y tabla 8.6 se detallan los costes para cada cargo de ambas planificaciones.

Cargo		Nº de días	Nº de horas/día	Nº de horas	Dedicación (hombre*mes)	Total (€)
Jefe de proyecto		10	3	30	0,33	472,49
Analista		30	3	90	1,00	1539,14
Investigador		18	3	54	0,60	950,85
Programador		99	3	297	3,30	3896,77
Documentador	Tareas	113	0,5	56,5	0,63	960,50
	Memoria	89	3	267	2,97	4539,00
<b>TOTAL</b>						<b>12358,75 €</b>

*Tabla 8.5 - Planificación inicial: Resumen de costes de personal*

Cargo	Nº de días	Nº de horas/día	Nº de horas	Dedicación (hombre*mes)	Total (€)
<b>Jefe de proyecto</b>	20	3	60	0,67	947,60
<b>Analista</b>	38	3	114	1,27	1954,97
<b>Investigador</b>	26	3	78	0,87	1377,25
<b>Programador</b>	278	3	834	9,27	10942,45
<b>Operador</b>	8	3	24	0,27	219,12
<b>Documentador</b>	<b>Tareas</b>	243	0,5	121,5	2065,50
	<b>Memorias</b>	109	3	327	5559,00
<b>TOTAL</b>					<b>23065,89 €</b>

*Tabla 8.6 - Planificación final: Resumen de costes de personal*

Para calcular la dedicación de cada uno de los cargos del proyecto se utiliza la ecuación 8.2. Cada jornada laboral es de 3 horas/día de modo que un hombre dedica 90 horas al mes al proyecto.

$$\text{Dedicación (hombre * mes)} = \frac{n^{\circ} \text{ horas dedicadas}}{\left( \frac{30 \text{ días/mes}}{\text{hombre}} \right) \left( \frac{3 \text{ horas/día}}{\text{hombre}} \right)} = \frac{n^{\circ} \text{ horas dedicadas}}{90 \frac{\text{horas}}{\text{hombre * mes}}}$$

*Ecuación 8.2 - Cálculo de la dedicación (hombre mes)*

El coste total de cada cargo se calcula con la dedicación y el salario mensual correspondiente a éste según se detalla en la ecuación 8.3. Por tanto el coste total de personal es la suma de los costes totales de cada uno de los cargos.

$$\text{Total (€)} = (\text{dedicación}) (\text{salario mensual})$$

*Ecuación 8.3 - Cálculo del salario mensual*

### 8.3.3 Equipos

En la tabla 8.7 y tabla 8.8 se detallan los costes de amortiguación de cada equipo utilizado para el desarrollo del proyecto de las dos planificaciones. Todos los costes son calculados sin I.V.A (21%).

Equipo	Coste con I.V.A (€)	Coste sin I.V.A (€)	% uso dedicado al proyecto	Dedicación (meses)	Coste imputable (€)
Portátil Asus A52J	699	552,21	100	9	82,83
Router Zyxel	20	15,8	100	9	2,37
smartphone Cubot One	198	156,42	100	9	23,46
<b>TOTAL</b>					<b>108,66 €</b>

*Tabla 8.7 - Planificación inicial: Resumen de coste de equipos*

Equipo	Coste con I.V.A (€)	Coste sin I.V.A (€)	% uso dedicado al proyecto	Dedicación (meses)	Coste imputable (€)
Portátil Asus A52J	699	552,21	100	17	156,46
Router Zyxel	20	15,8	100	17	4,48
smartphone Cubot One	198	156,42	100	17	44,32
<b>TOTAL</b>					<b>205,26 €</b>

*Tabla 8.8 - Planificación final: Resumen de coste de equipos*

Para realizar el cálculo de los costes imputables de los equipos necesarios para llevar a cabo el desarrollo del proyecto se utiliza la ecuación 8.4. De modo que el coste imputable total es la suma de los costes imputables de cada equipo individual.

$$\text{Amortización} = \frac{A}{B} * C * D$$

$A = n^{\circ}$  de meses desde la fecha de facturación

$B =$  periodo de depreciación (60 meses)

$C =$  coste del equipo (sin I.V.A)

$D =$  % de uso que se dedica al proyecto

*Ecuación 8.4 - Cálculo de la amortización*

### 8.3.4 Otros costes

En la tabla 8.9 y tabla 8.10 se detallan los costes del material (bolígrafos, folios, lápices, clips, grapadoras, carpetas, etc.) necesario para la realización del proyecto así como del transporte a la Universidad y de la tarifa de conexión a Internet para ambas planificaciones. Todos los costes son calculados sin I.V.A (21%).

Descripción	Coste con I.V.A (€)	Coste sin I.V.A (€)
Material de oficina	50	39,5
Conexión ADSL (*)	378	298,62
Transporte	30	23,7
<b>TOTAL</b>		<b>361,82 €</b>

*Tabla 8.9 - Resumen de otros costes directos inicial*

Descripción	Coste con I.V.A (€)	Coste sin I.V.A (€)
Material de oficina	50	39,5
Conexión ADSL (**)	714	564,06
Transporte	30	23,7
<b>TOTAL</b>		<b>627,26 €</b>

*Tabla 8.10 - Resumen de otros costes directos final*

(\*) La tarifa de la conexión ADSL es de 42€/mes (I.V.A incluido), por tanto, el coste de la conexión por 9 meses de duración del proyecto es de 378€.

(\*\*) El coste de la conexión ADSL por 17 meses de duración es de 714€.



### 8.3.5 Presupuesto total

En la tabla 8.11 y tabla 8.12 se muestran los costes totales (sin I.V.A) de los subapartados anteriores. Para calcular el presupuesto total del proyecto se añade al sumatorio de los costes totales anteriores un veinte por ciento en concepto de costes indirectos.

Costes	Presupuesto costes totales (€)
Personal	12358,75
Amortización	108,66
Otros costes	361,82
Costes indirectos (20%)	2565,85
<b>TOTAL</b>	<b>15395,08 €</b>

*Tabla 8.11 - Planificación inicial: Resumen del presupuesto sin I.V.A*

Costes	Presupuesto costes totales (€)
Personal	23065,89
Amortización	205,26
Otros costes	627,26
Costes indirectos (20%)	4779,45
<b>TOTAL</b>	<b>28678,09 €</b>

*Tabla 8.12 - Planificación final: Resumen del presupuesto sin I.V.A*

El presupuesto total de este proyecto en la planificación inicial asciende a la cantidad de QUINCE MIL TRESCIENTOS NOVENTA Y CINCO EUROS CON OCHO CÉNTIMOS. Sin embargo, el presupuesto total en la planificación final asciende a la cantidad de VEINTIOCHO MIL SEISCIENTOS SETENTA Y OCHO EUROS CON NUEVE CÉNTIMOS.

La diferencia entre ambos es del total de (13283,01 €) TRECE MIL DOSCIENTOS OCHENTA Y TRES EUROS CON UN CÉNTIMO. En un proyecto real, dicha diferencia sería comunicada previamente al cliente.

### 8.3.6 Plantilla resumen

#### 8.3.6.1 Plantilla presupuesto inicial

En la figura 8.6 se muestran los costes estimados iniciales para la realización del proyecto.

**PRESUPUESTO DE PROYECTO****1.- Autor**

Yolanda Budia Escribano

**2.- Departamento**

Sistemas y Automática

**3.- Descripción del proyecto**

- **Título:** Interfaz para teleoperar un robot social desde móvil
- **Duración (meses):** 9
- **Tasa de costes indirectos:** 20%

**4.- Presupuesto total del proyecto (valores en Euros)**

15395,08 Euros

**5.- Desglose presupuestario (costes directos)****PERSONAL**

Apellidos y nombre	Cargo	Dedicación (hombre mes) (*)	Coste hombre mes (€)	Coste (€)	
Budia Escribano, Yolanda	Jefe de proyecto	0,33	1417,47	472,49	
Budia Escribano, Yolanda	Analista	1,00	1539,14	1539,14	
Budia Escribano, Yolanda	Investigador	0,60	1584,75	950,85	
Budia Escribano, Yolanda	Programador	3,30	1180,84	3896,77	
Budia Escribano, Yolanda	Documentador	Tareas	0,63	1530,00	960,5
		Memoria	2,97	1530,00	4539,00
			TOTAL	12358,75 €	

\* 1 hombre mes = 90 horas según la planificación realizada.



## EQUIPOS

Descripción	Coste sin I.V.A (€)	% uso dedicado al proyecto	Dedicación (meses)	Periodo de depreciación (meses)	Coste imputable (€) (**)
Portátil Asus A52J	552,21	100	9	60	82,83
Router Zyxel	15,8	100	9	60	2,37
smartphone Cubot One	156,42	100	9	60	23,46
				<b>TOTAL</b>	<b>108,66 €</b>

\*\* Fórmula de cálculo de la Amortización  $\frac{A}{B} * C * D$

A = nº de meses desde la fecha de facturación

B = periodo de depreciación (60 meses)

C = coste del equipo (sin I.V.A)

D = % de uso que se dedica al proyecto

## OTROS COSTES DIRECTOS DEL PROYECTO

Descripción	Coste sin I.V.A (€)
Material de oficina	39,5
Conexión ADSL	298,62
Transporte	23,7
<b>TOTAL</b>	<b>361,82 €</b>

## 6.- Resumen de costes

Costes	Presupuesto costes totales (€)
Personal	12358,75
Amortización	108,66
Otros costes	361,82
Costes indirectos (20%)	2565,85
<b>TOTAL</b>	<b>15395,08 €</b>

Figura 8.6 - Plantilla de presupuesto inicial

### 8.3.6.2 Plantilla del presupuesto final

En la figura 8.7 se muestran los costes finales necesarios para la realización del proyecto.

PRESUPUESTO DE PROYECTO					
1.- Autor					
Yolanda Budia Escribano					
2.- Departamento					
Sistemas y Automática					
3.- Descripción del proyecto					
- Título:		Interfaz para teleoperar un robot social desde móvil			
- Duración (meses):		17			
- Tasa de costes indirectos:		20%			
4.- Presupuesto total del proyecto (valores en Euros)					
28678,09 Euros					
5.- Desglose presupuestario (costes directos)					
PERSONAL					
Apellidos y nombre	Cargo	Dedicación (hombre mes) (*)	Coste hombre mes (€)	Coste (€)	
Budia Escribano, Yolanda	Jefe de proyecto	0,67	1421,39	947,60	
Budia Escribano, Yolanda	Analista	1,27	1543,40	1954,97	
Budia Escribano, Yolanda	Investigador	0,87	1589,14	1377,25	
Budia Escribano, Yolanda	Programador	9,27	1180,84	10942,45	
Budia Escribano, Yolanda	Operador	0,27	821,69	219,12	
Budia Escribano, Yolanda	Documentador	Tareas	1,35	1530,00	2065,00
		Memoria	3,63	1530,00	5559,00
			TOTAL	23065,89 €	

\* 1 hombre mes = 90 horas según la planificación realizada.

## EQUIPOS

Descripción	Coste sin I.V.A (€)	% uso dedicado al proyecto	Dedicación (meses)	Periodo de depreciación (meses)	Coste imputable (€) (**)
Portátil Asus A52J	552,21	100	17	60	156,46
Router Zyxel	15,8	100	17	60	4,48
smartphone Cubot One	156,42	100	17	60	44,32
<b>TOTAL</b>					<b>205,26 €</b>

\*\* Fórmula de cálculo de la Amortización  $\frac{A}{B} * C * D$

A = nº de meses desde la fecha de facturación

B = periodo de depreciación (60 meses)

C = coste del equipo (sin I.V.A)

D = % de uso que se dedica al proyecto

## OTROS COSTES DIRECTOS DEL PROYECTO

Descripción	Coste sin I.V.A (€)
Material de oficina	39,50
Conexión ADSL	564,06
Transporte	23,70
<b>TOTAL</b>	<b>627,26 €</b>

## 6.- Resumen de costes

Costes	Presupuesto costes totales (€)
Personal	23065,89
Amortización	205,26
Otros costes	627,26
Costes indirectos (20%)	4779,68
<b>TOTAL</b>	<b>28678,09 €</b>

Figura 8.7 - Plantilla de presupuesto final



# Capítulo 9

## Conclusiones y líneas futuras

Para finalizar, se exponen las conclusiones obtenidas tras la realización del proyecto valorando los objetivos que se han cumplido. También se detallan las posibles líneas futuras que se pueden llevar a cabo para mejorar el proyecto realizado.

## 9.1 Conclusiones

La finalidad fundamental para el desarrollo del proyecto es diseñar e implementar una interfaz de usuario, que permita teleoperar un robot social mediante un dispositivo móvil. Con la realización de este proyecto se ha querido conseguir el objetivo social de mejorar la calidad de vida de los cuidadores de personas con demencia con sólo manejar un *smartphone*, que a día de hoy tiene la mayoría de la población.

Para lograr los objetivos que se marcaron inicialmente, se han implementado tres versiones de la aplicación para teleoperar el robot. Todas ellas tienen en común la capacidad de poder ser utilizadas por el mayor número de usuarios de cualquier edad y nivel de conocimientos académicos y/o técnicos. Conseguir esta capacidad ha resultado un trabajo bastante laborioso dentro de este proyecto, puesto que el estudio más importante a realizar ha sido comparar las diferentes características de las aplicaciones móviles disponibles en la actualidad, y elegir la más adecuada para poder cumplir con todos los requisitos impuestos al inicio.

Tras valorar las diferentes opciones disponibles, se ha decidido desarrollar una aplicación web debido a que permite crear una interfaz de usuario con las funcionalidades requeridas, compatible con cualquier sistema operativo evitando así, tener que desarrollar una aplicación diferente para cada plataforma. Cabe destacar, que la comunicación con ROS también ha intervenido en la decisión para elegir el tipo de aplicación, puesto que *rosjava* se encuentra aún en fase experimental mientras que la librería *roslibjs* de *rosbridge* permite realizar dicha comunicación de una manera sencilla y estable mediante el uso de *WebSockets*.

La realización del proyecto ha sufrido una variación en la planificación temporal por la incorporación de mejoras en la aplicación. En la planificación inicial se impuso una fecha límite para finalizar el proyecto, de modo que se desarrolló la primera versión de manera rápida y que permitiera cumplir con los requisitos iniciales. Sin embargo, para mejorar la aplicación añadiendo persistencia a las tareas ha sido necesario el nodo ROS “*activity\_persistence\_module*” que contiene los servicios “*request\_activity*” y “*current\_activity*” y el tema “*activity\_finished*” para lanzar tareas, consultar el estado del robot y actualizar el estado cuando una tarea finaliza, respectivamente. Puesto que *Mini* es un robot en desarrollo, no tenía implementado dicho nodo, por lo que se ha diseñado e implementado en este proyecto retrasando la fecha de entrega del mismo. Además dicho módulo puede integrarse en el control central del robot en el futuro para usarlo en otras aplicaciones.

Una vez finalizada la aplicación web v.1.0 se ha evaluado con usuarios reales con edades y niveles de conocimiento y estudios diferentes, de modo que se puede confirmar que la interfaz de teleoperación es sencilla e intuitiva. No obstante, para incorporar las mejoras sugeridas por los usuarios en la evaluación, incrementando la usabilidad de la aplicación y para poder añadir más funcionalidades de otros escenarios en trabajos futuros, se ha rediseñado la aplicación v.1.0 implementando una segunda versión web capaz de pasar a aplicación híbrida para acceder al hardware del dispositivo.

Las tres versiones implementadas tienen características diferentes pero en todas se mantiene la funcionalidad de la aplicación. Mediante el uso de dichas aplicaciones el usuario es capaz de consultar la guía de ayuda, cambiar el idioma de la interfaz, realizar

búsquedas de tareas, seleccionar una funcionalidad, seleccionar una tarea, solicitar al robot que realice, pare, pause o reanude una tarea y consultar el estado del robot.

El resumen de las características de cada versión se describe a continuación:

- Aplicación web v.1.0: es una aplicación multiplataforma, cuyo código debe estar en el lado del servidor, y es necesaria la recarga completa de la página. No requiere ningún tipo de instalación en el dispositivo, únicamente un navegador web instalado en éste, por lo que no puede acceder al hardware del dispositivo. Esta versión ha sido evaluada por usuarios con diferentes perfiles.
- Aplicación web v.2.0: es una aplicación multiplataforma, cuyo código se divide en dos, la parte del cliente y la del servidor, de manera que se puede ejecutar la aplicación desde un equipo remoto al servidor accediendo a éste cuando sea necesario, eliminando la recarga completa de la página. No requiere ningún tipo de instalación en el dispositivo, únicamente un navegador web instalado en éste, por lo que no puede acceder al hardware del dispositivo. Además, incorpora las mejoras sugeridas por los usuarios tras la evaluación de la primera versión.
- Aplicación híbrida v.2.1: se trata de la aplicación v.2.0 adaptada a híbrida mediante Phonegap/Cordova para el sistema operativo Android, que permite acceder a parte del hardware del dispositivo. Requiere instalación en el dispositivo, de manera que no es necesario el navegador web, mejorando el acceso a la aplicación. Sin embargo, el mantenimiento no es inmediato, es necesario reinstalar la aplicación para futuras actualizaciones.

Como conclusión final, se puede confirmar que las tres versiones cumplen con los requisitos y objetivos iniciales del proyecto:

- Interfaz sencilla e intuitiva para usuarios de cualquier edad y nivel de conocimientos académicos y/o técnicos.
- Comunicación entre la aplicación y el robot a través de ROS. Permitir el envío de tareas desde la aplicación para ser realizadas por el robot.
- Teleoperar el robot social desde un dispositivo móvil de manera remota a través de Internet, por medio de conexión WiFi o datos móviles que tenga contratado el usuario.
- Almacenamiento y consulta de los datos de la base de datos proporcionada.
- Uso de la aplicación por el mayor número de usuarios.
- Adaptación a cualquier tamaño de pantalla.

La realización de este proyecto ha servido al proyectista a:

- Descubrir las diversas funcionalidades que pueden realizar los robots sociales para mejorar la calidad de vida de la sociedad.
- Estudiar las posibilidades que ofrecen las aplicaciones web.
- Descubrir ROS y aprender a comunicar una aplicación web con un robot mediante *rosbridge*.
- Aprender a utilizar los *frameworks* Bootstrap y Phonegap/Cordova.
- Aprender a utilizar Eclipse, Android Studio y PHPMyAdmin.

- Aprender las tecnologías HTML, CSS, JavaScript, PHP, AJAX, C++, MySQL y MySQL++.

No obstante, también se han encontrado las siguientes dificultades que se han ido solventando exitosamente en el transcurso del proyecto:

- Base de datos:

La información relevante para la aplicación se encuentra almacenada en diferentes tablas y vistas de la base de datos proporcionada, de modo que ha sido necesario aprender a crear vistas y tablas MySQL para guardar dicha información.

- ROS:

Inicialmente se iban a proporcionar los servicios y temas necesarios para la comunicación entre la aplicación y el robot. Sin embargo, para poner añadir persistencia a las tareas, se ha decidido implementar el nodo “activity\_persistence\_module” que no estaba implementado y era necesario para el uso de la aplicación.

Al ser un robot en pleno desarrollo, cada servicio y tema de ROS de éste está realizado por una persona del equipo diferente, por lo que algunos de los temas necesarios no estaba finalizado y hubo que adaptarlo.

- Funcionalidades de la aplicación:

Al principio se puso como requisito que la aplicación contuviera las funcionalidades: música, fotos, tv/videos, noticias, historias y multimedia. Una vez iniciada la fase de implementación se propuso incluir otra funcionalidad que pudiera enviar notificaciones al dispositivo móvil. Por esta razón y con el fin de incorporar mejoras que pudieran acceder al hardware del dispositivo, se desarrolló la segunda versión de la aplicación que permitiera hacer el traspaso a aplicación híbrida (v.2.1).

## 9.2 Líneas futuras

Como aportación final al proyecto en cuestión, se han analizado las posibles mejoras que se podrían incorporar a la aplicación en trabajos futuros. Debido a que la segunda versión de la aplicación ha permitido reducir las limitaciones de la primera, las líneas futuras propuestas en este apartado se basan en la segunda y tercera versión de la aplicación.



- Añadir más idiomas:

Dado que la aplicación está implementada en inglés y español, se podrían incorporar más idiomas para que pueda ser utilizada por un mayor número de usuarios.

- Sesiones de usuario:

Crear sesiones de usuario para que cada uno pueda cambiar sus preferencias en la aplicación (idioma, colores, etc.).

- Servicios ROS:

Integración de los servicios y temas ROS desarrollados en este proyecto con el resto de habilidades que tiene el robot, de tal forma que éstas publiquen un mensaje en el tema “activity\_finished” cuando finalice la tarea que está realizando el robot, para que el estado pase a ser que no está realizando ninguna tarea en la aplicación. Dicha integración se ha probado con la funcionalidad historias.

- Base de datos:

Añadir la posibilidad de que el usuario, desde la aplicación, pueda insertar, editar o eliminar elementos de la lista de tareas disponibles para cada funcionalidad, con la finalidad de personalizar el contenido de la base de datos a los gustos del paciente.

- Añadir más funcionalidades a la aplicación:

1. Funcionalidad juegos: actualmente el robot es capaz de realizar juegos, por lo que se podría incorporar la funcionalidad de juegos a la aplicación para que el cuidador pueda elegir el juego que desea que realice el robot con el paciente.
2. Enviar avisos de emergencia a la aplicación: si el robot detecta que el anciano se ha levantado o caído enviará un aviso a la aplicación. El dispositivo móvil recibirá una notificación con el aviso para que lo vea el cuidador. Esta mejora sólo se puede implementar en la aplicación híbrida porque es necesario acceder al hardware del dispositivo.
3. Llamar a emergencias: al igual que la funcionalidad anterior, cuando el robot envíe el aviso a la aplicación, ésta realizará una llamada a la persona indicada (a emergencias o a algún familiar).
4. Toma de fotografías: la aplicación solicita al robot que le envíe una foto actual de la persona mayor, el robot hará una foto mediante la *Kinect* que tiene incorporada y la enviará a la aplicación.
5. Crear una aplicación híbrida para el sistema operativo móvil iOS, debido a que es el que sigue a Android en la cuota de mercado, para que un mayor número de usuarios pueda hacer uso de las funcionalidades anteriores descritas.



# **Anexo A**

## **Encuesta**

En este anexo se muestra la encuesta realizada con diversos tipos de usuarios para evaluar y probar la aplicación con el fin de mejorar las características y funcionalidades de ésta. Se realizan preguntas de la mayoría de las funcionalidades para hacer un estudio más exhaustivo de los resultados.

## A.A Perfil de usuario

### A1.- Rango de edad

- ☐ < 16
- ☐ 16 – 19
- ☐ 20 - 29
- ☐ 30 - 39
- ☐ 40 – 50
- ☐ > 50

### A2.- Nivel de formación

- ☐ Estudios primarios
- ☐ Bachillerato
- ☐ Formación profesional
- ☐ Grado
- ☐ Postgrado / Máster
- ☐ Doctorado
- ☐ Otro

### A3.- ¿Cuál es tu formación?

Ej: telecomunicación, industrial, informática, educación, administración...

### A4.- Nivel de experiencia con *smartphones*

- ☐ Ninguno
- ☐ Básico
- ☐ Normal
- ☐ Bueno
- ☐ Experto

### A5.- Indica el modelo de *smartphone* utilizado

Ej: Sony Experia Z, HTC One, Samsung Galaxy S4, ...

### A6.- ¿Conocía previamente el robot?

- ☐ No lo había visto nunca
- ☐ Sólo conoce alguna función que realiza
- ☐ Conoce todo lo que puede realizar el robot
- ☐ Había hecho pruebas anteriormente con él
- ☐ Sabe manejarlo perfectamente
- ☐ Otro

### A7.- Comentarios adicionales sobre el perfil

## A.B Usabilidad de la interfaz

### B1.- ¿La funcionalidad de cada botón le ha parecido clara?

El 1 indica muy poco y el 5 mucho

1 2 3 4 5  
☐ ☐ ☐ ☐ ☐

### B2.- ¿La selección de colores le parece adecuada?

El 1 indica muy poco y el 5 mucho

1 2 3 4 5  
☐ ☐ ☐ ☐ ☐

### B3.- ¿El contenido del menú es suficiente?

- ☐ Le faltan elementos
- ☐ Debería contener sólo la barra de búsqueda, el cambio de idioma y la ayuda
- ☐ Debería contener sólo las funcionalidades del robot
- ☐ Contiene todo lo necesario
- ☐ Otro

### B4.- Sugerencias sobre el menú

Indique qué cambios considera necesarios para mejorar el menú

### B5.- ¿Cree conveniente la aparición del CUADRO DE CONFIRMACIÓN al seleccionar una actividad?

El 1 indica muy poco y el 5 mucho

1 2 3 4 5  
☐ ☐ ☐ ☐ ☐

### B6.- ¿La información del CUADRO DE CONFIRMACIÓN le parece suficiente y clara?

Puede seleccionar múltiples opciones

- ☐ No, no es insuficiente
- ☐ No, no se entiende bien
- ☐ Sí, está clara
- ☐ Sí, es suficiente
- ☐ Otro

### B7.- ¿Cree conveniente la aparición del CUADRO DE INFORMACIÓN de reproducción?

El 1 indica muy poco y el 5 mucho

1 2 3 4 5  
☐ ☐ ☐ ☐ ☐

**B8.- ¿La información del CUADRO DE INFORMACIÓN le parece suficiente y clara?**

Puede seleccionar múltiples opciones

- ☐ No, no es suficiente
- ☐ No, no se entiende bien
- ☐ Sí, está clara
- ☐ Sí, es suficiente
- ☐ Otro

**B9.- Sugerencias sobre los cuadros de confirmación y/o de información**

Indique qué cambios considera necesarios para mejorar la interfaz.

**B10.- ¿La interfaz le ha parecido intuitiva y fácil de manejar?**

El 1 indica muy poco y el 5 mucho

- |                       |                       |                       |                       |                       |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1                     | 2                     | 3                     | 4                     | 5                     |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

**B11.- Sugerencias sobre la apariencia de la interfaz**

Indique qué cambios considera necesarios para mejorar la interfaz.

## A.C Actividad "Ver ayuda"

**C1.- ¿El acceso a la ayuda le ha resultado sencillo?**

El 1 indica muy poco y el 5 mucho

- |                       |                       |                       |                       |                       |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1                     | 2                     | 3                     | 4                     | 5                     |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

**C2.- ¿La ayuda proporcionada en la interfaz es clara?**

El 1 indica muy poco y el 5 mucho

- |                       |                       |                       |                       |                       |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1                     | 2                     | 3                     | 4                     | 5                     |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

**C3.- ¿La ayuda proporcionada en la interfaz le ha parecido suficiente?**

- ☐ Faltan cosas por explicar
- ☐ Es muy corta
- ☐ Es muy extensa
- ☐ Contiene lo necesario para poder utilizar correctamente la interfaz
- ☐ Otro

**C4.- Sugerencias sobre el contenido de la ayuda**

Indique qué cambios considera necesarios para mejorar la ayuda.

## A.D Actividad "Cambiar de idioma"

**D1.- ¿El acceso al cambio de idioma le ha resultado sencillo?**

El 1 indica muy poco y el 5 mucho

- 1   2   3   4   5  
☐   ☐   ☐   ☐   ☐

**D2.- ¿El idioma de la interfaz se corresponde con el idioma elegido?**

- ☐ No ha cambiado de idioma
- ☐ El idioma de la interfaz se ha cambiado a un idioma distinto del elegido
- ☐ Se ha cambiado al idioma elegido
- ☐ Otro

**D3.- Sugerencias sobre la selección del idioma**

Indique qué cambios considera necesarios para mejorar la selección del idioma.

## A.E Actividad "Consultar el estado del robot"

**E1.- ¿La consulta del estado del robot le ha resultado sencilla?**

El 1 indica muy poco y el 5 mucho

- 1   2   3   4   5  
☐   ☐   ☐   ☐   ☐

**E2.- ¿La información que se muestra en el estado es suficiente?**

El 1 indica muy poco y el 5 mucho

- 1   2   3   4   5  
☐   ☐   ☐   ☐   ☐

**E3.- ¿El estado del robot se corresponde con la actividad que está realizando?**

En caso de respuesta negativa, explique los motivos

- ☐ Sí
- ☐ No

Motivos

**E4.- Sugerencias sobre la consulta del estado**

Indique qué cambios considera necesarios para mejorar la consulta del estado.

## A.F Actividad "Ver lista de música"

**F1.- ¿La página de lista de música le parece intuitiva?**

El 1 indica muy poco y el 5 mucho

1   2   3   4   5  
☐   ☐   ☐   ☐   ☐

**F2.- ¿La reproducción de una canción le ha resultado sencilla?**

El 1 indica muy poco y el 5 mucho

1   2   3   4   5  
☐   ☐   ☐   ☐   ☐

**F3.- ¿El comportamiento se ha correspondido con el esperado?**

En caso de respuesta negativa, explique los motivos

- ☐ Sí
- ☐ No

Motivos

**F4.- ¿Al presionar el botón de Stop la canción se ha parado?**

En caso de respuesta negativa, explique los motivos

- ☐ Sí
- ☐ No

Motivos

**F5.- Sugerencias acerca de la lista de música**

Indique qué cambios considera necesarios para mejorar en la lista de música

## A.G Actividad "Ver lista de canales de Tv/Videos"

**G1.- ¿La página de lista de canales le parece intuitiva?**

El 1 indica muy poco y el 5 mucho

1   2   3   4   5  
☐   ☐   ☐   ☐   ☐



**G2.- ¿La visualización de un canal le ha resultado sencilla?**

El 1 indica muy poco y el 5 mucho

- |                       |                       |                       |                       |                       |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1                     | 2                     | 3                     | 4                     | 5                     |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

**G3.- ¿El comportamiento se ha correspondido con el esperado?**

En caso de respuesta negativa, explique los motivos

- ☐ Sí
- ☐ No

Motivos

**G4.- ¿Al presionar el botón de Stop el video/canal se ha parado?**

En caso de respuesta negativa, explique los motivos

- ☐ Sí
- ☐ No

Motivos

**G5.- Sugerencias acerca de la lista de canales**

Indique qué cambios considera necesarios para mejorar en la lista de canales

## A.H Actividad "Ver lista de historias"

**H1.- ¿La página de lista de historias le parece intuitiva?**

El 1 indica muy poco y el 5 mucho

- |                       |                       |                       |                       |                       |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1                     | 2                     | 3                     | 4                     | 5                     |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

**H2.- ¿La reproducción de una historia le ha resultado sencilla?**

El 1 indica muy poco y el 5 mucho

- |                       |                       |                       |                       |                       |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1                     | 2                     | 3                     | 4                     | 5                     |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

**H3.- ¿El comportamiento se ha correspondido con el esperado?**

En caso de respuesta negativa, explique los motivos

- ☐ Sí
- ☐ No

Motivos

**H4.- ¿Al presionar el botón de Play/Pause la historia se ha reanudado/pausado?**

En caso de respuesta negativa, explique los motivos

- ☐ Sí
- ☐ No

Motivos

**H5.- ¿Al presionar el botón de Stop la historia se ha parado?**

En caso de respuesta negativa, explique los motivos

- ☐ Sí
- ☐ No

Motivos

**H6.- Sugerencias acerca de la lista de historias**

Indique qué cambios considera necesarios para mejorar en la lista de historias

## **A.I Actividad "Ver lista de contenidos multimedia"**

**I1.- ¿La página de lista de multimedia le parece intuitiva?**

El 1 indica muy poco y el 5 mucho

- |                       |                       |                       |                       |                       |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1                     | 2                     | 3                     | 4                     | 5                     |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

**I2.- ¿La reproducción del contenido relacionado con una etiqueta le ha resultado sencilla?**

El 1 indica muy poco y el 5 mucho

- |                       |                       |                       |                       |                       |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1                     | 2                     | 3                     | 4                     | 5                     |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

**I3.- ¿La selección de múltiples etiquetas le ha resultado sencilla?**

El 1 indica muy poco y el 5 mucho

- |                       |                       |                       |                       |                       |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1                     | 2                     | 3                     | 4                     | 5                     |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

**I4.- ¿El comportamiento se ha correspondido con el esperado?**

En caso de respuesta negativa, explique los motivos

- ☐ Sí
- ☐ No

Motivos

**I5.- ¿Al presionar el botón de Pause la reproducción multimedia se ha pausado?**

En caso de respuesta negativa, explique los motivos

- ☐ Sí
- ☐ No

Motivos

**I6.- ¿Al presionar el botón de Play la reproducción multimedia se ha reanudado?**

En caso de respuesta negativa, explique los motivos

- ☐ Sí
- ☐ No

Motivos

**I7.- ¿Al presionar el botón de Stop la reproducción multimedia se ha parado?**

En caso de respuesta negativa, explique los motivos

- ☐ Sí
- ☐ No

Motivos

**I8.- Sugerencias acerca de la lista de contenidos multimedia**

Indique qué cambios considera necesarios para mejorar en la lista de contenidos multimedia

## A.J Realizar búsqueda

**J1.- ¿El acceso a la búsqueda le parece intuitivo?**

El 1 indica muy poco y el 5 mucho

1   2   3   4   5  
☐   ☐   ☐   ☐   ☐

**J2.- ¿Le ha resultado fácil buscar en todas las funcionalidades?**

El 1 indica muy poco y el 5 mucho

1   2   3   4   5  
☐   ☐   ☐   ☐   ☐

**J3.- ¿Le ha resultado fácil buscar en UNA funcionalidad?**

El 1 indica muy poco y el 5 mucho

1   2   3   4   5  
☐   ☐   ☐   ☐   ☐

**J4.- ¿La búsqueda ha sido correcta?**

En caso negativo indique el motivo

- ☐ Sí
- ☐ No

Motivos

## A.K Experiencia general

**K1.- ¿La interfaz ha cumplido con sus expectativas iniciales?**

- ☐ La interfaz no es lo que esperaba
- ☐ La interfaz NO permite teleoperar las actividades deseadas
- ☐ La interfaz permite teleoperar más actividades de las que esperaba
- ☐ La interfaz cumple con lo necesario
- ☐ Otro

**K2.- Comentarios para mejorar la interfaz**

# **Anexo B**

## **Manual de usuario**

En este manual se describen y se muestran las características correspondientes a las versiones 2.0 y 2.1 (aplicación web y aplicación híbrida respectivamente) de la interfaz de teleoperación desarrollada. El objetivo de este manual es enseñar al usuario a utilizar la aplicación, a comprender sus funcionalidades y a experimentar con todas sus posibilidades.

Después de su lectura, el usuario será capaz de cambiar de idioma, buscar, seleccionar, parar, pausar y reanudar una tarea, así como de consultar el estado del robot.

## B.1 Acceder a la aplicación

### APLICACIÓN WEB

Para iniciar la aplicación “teleopMobile” se requiere de un navegador:

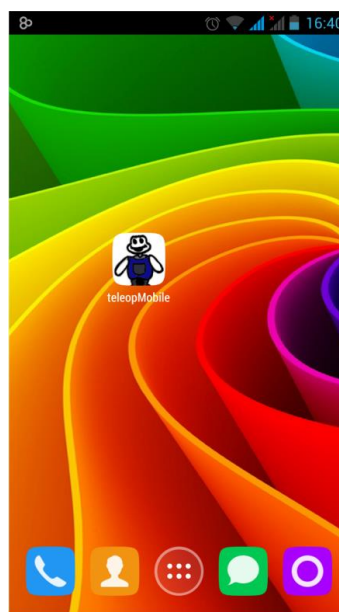
- 1- Iniciar la aplicación navegador del dispositivo móvil. En este caso se va a utilizar el navegador Chrome.



### APLICACIÓN HÍBRIDA

Para iniciar la aplicación “teleopMobile” se requiere:

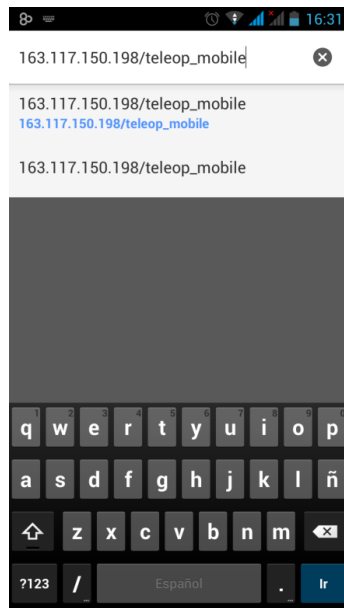
- 1- Descargar la aplicación “teleopMobile.apk”.
- 2- Instalarla en el dispositivo móvil.
- 3- Iniciar la aplicación pulsando sobre el icono de ésta.



*Figura B.1 - Acceso a la aplicación web e híbrida*

- 1- Introduzca la URL correspondiente en la barra de direcciones (figura B.2):  
*163.117.150.198/teleop\_mobile (\*)*
- 2- Pulse “Ir” en el teclado táctil de su dispositivo.

(\*) El número 163.117.150.198 hace referencia a la IP del robot o del router con dirección pública que le dé acceso. Se debe modificar dicha dirección si cambia o se le asigna un dominio al robot.



*Figura B.2 - Insertar URL en la aplicación web*

## B.2 Seleccionar idioma

Al iniciar la aplicación le aparecerá la página de selección de idioma (figura B.3). Actualmente la aplicación está disponible en dos idiomas: inglés y español. Seleccione una de las dos banderas para que el idioma de la aplicación se adapte a sus preferencias.



*Figura B.3 - Página de selección de idioma*

## B.3 Consultar estado del robot

Una vez seleccionado el idioma deseado, se muestra la página principal de la aplicación (figura B.4) en la cual se puede consultar el estado y/o seleccionar una funcionalidad. En este ejemplo se ha seleccionado el idioma español.

En el cuadro verde de la parte superior (remarcado en rojo) se puede consultar el estado del robot. Se puede acceder a esta página desde cualquier otra seleccionando “Inicio/Estado” de la barra de menú.



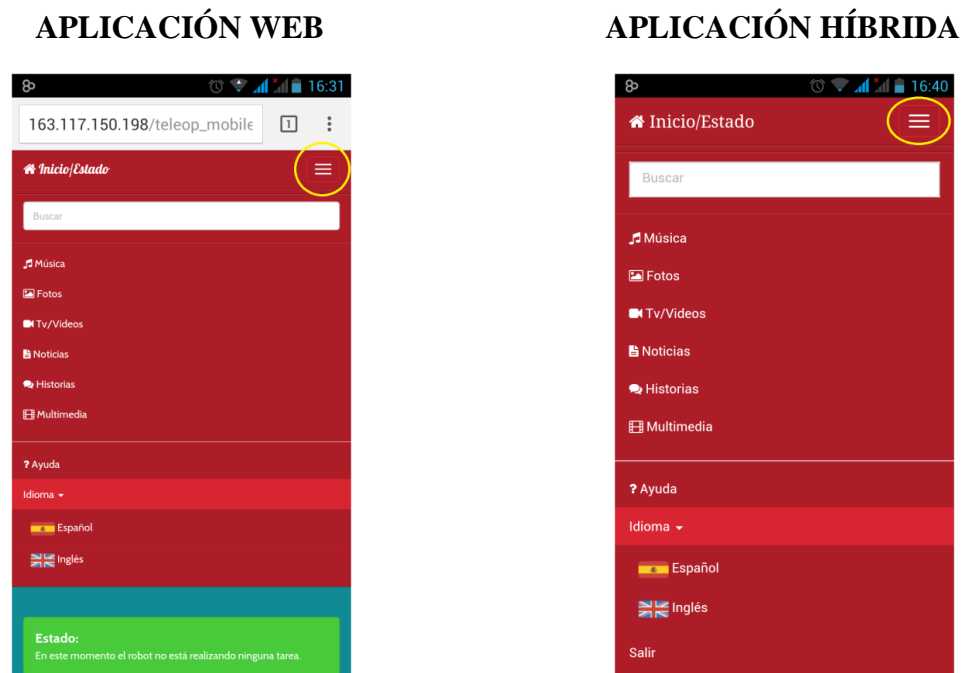
*Figura B.4 - Página principal/estado*

## B.4 Menú

Al pulsar el botón de la barra de menú con el icono de tres líneas horizontales (remarcado con un círculo amarillo), se desplegará el menú (figura B.5).

Desde cualquier página se puede acceder al menú y permite consultar el estado, cambiar de funcionalidad, realizar búsqueda de tareas, cambiar de idioma, consultar la guía de ayuda y salir de la aplicación (la última opción es únicamente para la híbrida).





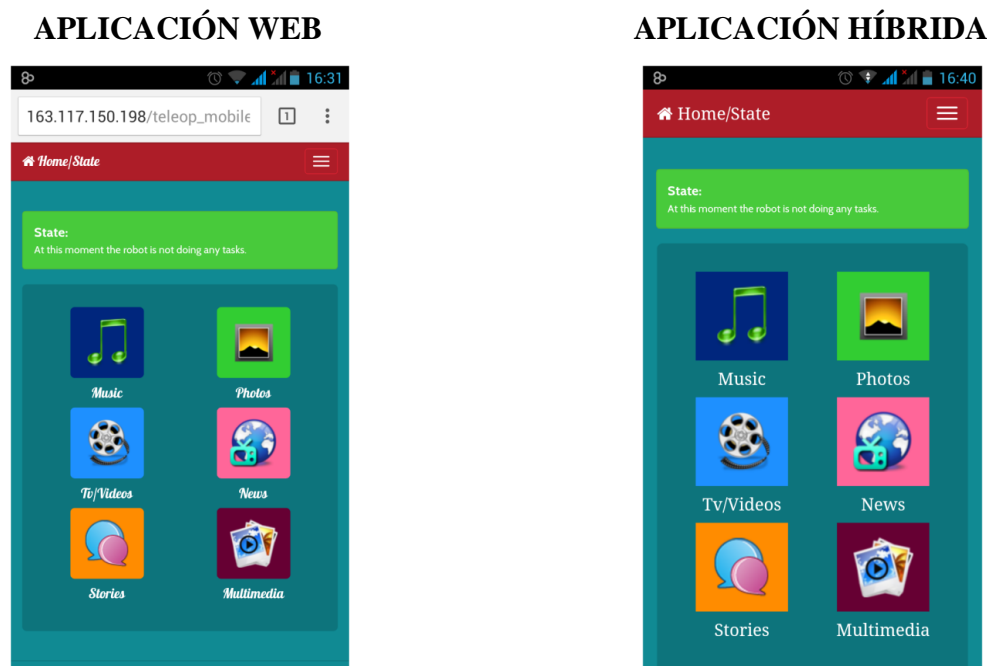
*Figura B.5 - Menú desplegado*

## B.5 Cambiar idioma

Una vez iniciada la aplicación es posible cambiar a otro idioma. Siga los siguientes pasos:

- 1- Despliegue el menú
- 2- Seleccione “Idioma”
- 3- En el siguiente desplegable, escoja el idioma deseado.

A continuación la aplicación se recargará con el idioma elegido (Ejemplo al seleccionar inglés: figura B.6).



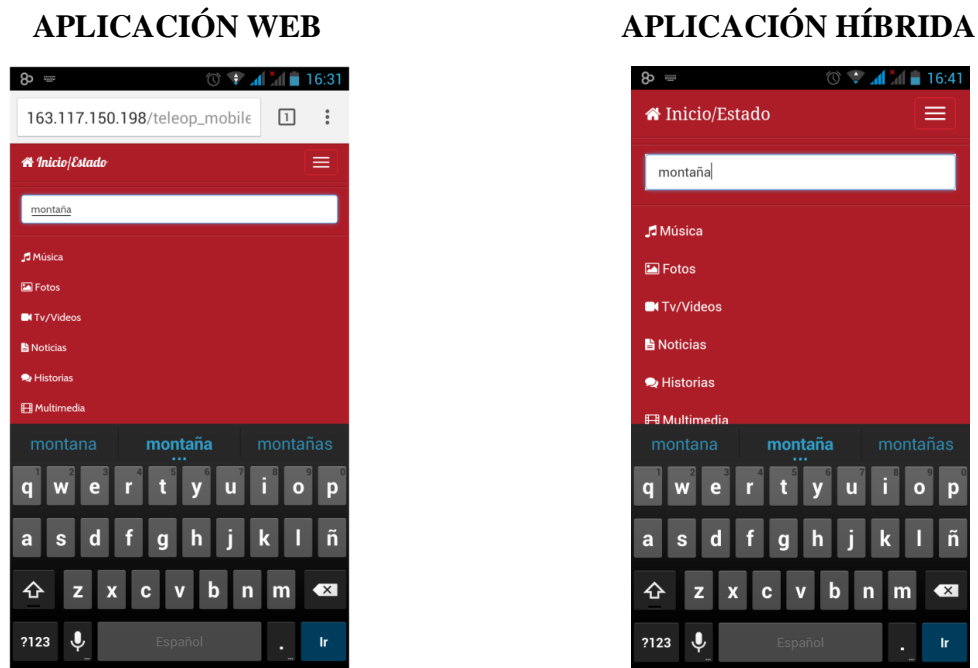
*Figura B.6 - Página principal/estado en otro idioma*

## B.6 Buscar tarea

La aplicación permite buscar tareas de todas las funcionalidades. La búsqueda no es *case sensitive*, es decir, no distingue entre mayúsculas, minúsculas ni tildes.

Para realizar una búsqueda siga los siguientes pasos:

- 1- Despliegue el menú.
- 2- Seleccione la barra de búsqueda.
- 3- Introduzca el texto de la tarea a buscar (en este ejemplo buscaremos las tareas que contengan la palabra *montaña*, figura B.7).
- 4- Seleccione la tecla “Ir” del teclado táctil del dispositivo.



*Figura B.7 - Búsqueda de una tarea*

La aplicación mostrará una página con el resultado de la búsqueda (figura B.8). Podrá seleccionar la tarea deseada para que el robot la realice.

Por defecto, aparecen ordenados alfabéticamente por título. Se puede cambiar el orden pulsando sobre las flechas (▲ o ▼) que hay debajo de los títulos “TIPO”, “TÍTULO” o “DESCRIPCIÓN” para ordenar por tipo, títulos o descripciones respectivamente.



*Figura B.8 - Página de resultado de la búsqueda*

## B.7 Funcionalidades

Al seleccionar una de las funcionalidades se mostrará la página con la lista de tareas disponibles para dicha funcionalidad.

Por defecto, aparecen ordenados alfabéticamente por título. Se puede cambiar el orden pulsando sobre las flechas (▲ o ▼) que hay debajo de los títulos “TÍTULO” Y “DESCRIPCIÓN” para ordenar por títulos o descripciones respectivamente.

### B.7.1 Música

Al seleccionar la funcionalidad música, aparecerá la página con la lista de canciones disponibles (figura B.9).

Para que el robot deje de reproducir la canción pulse el botón rojo “Parar”.

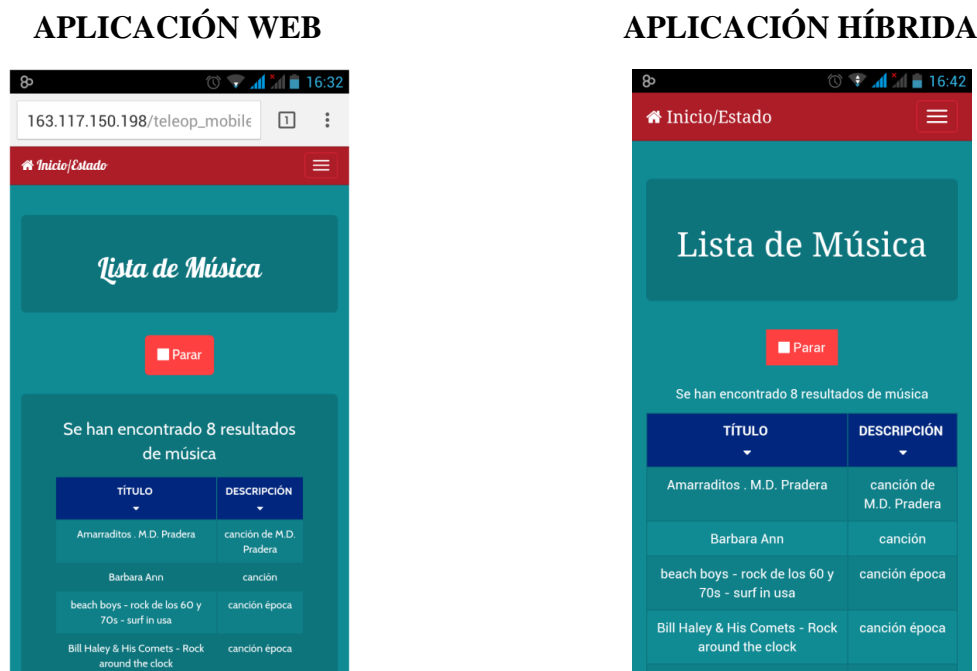
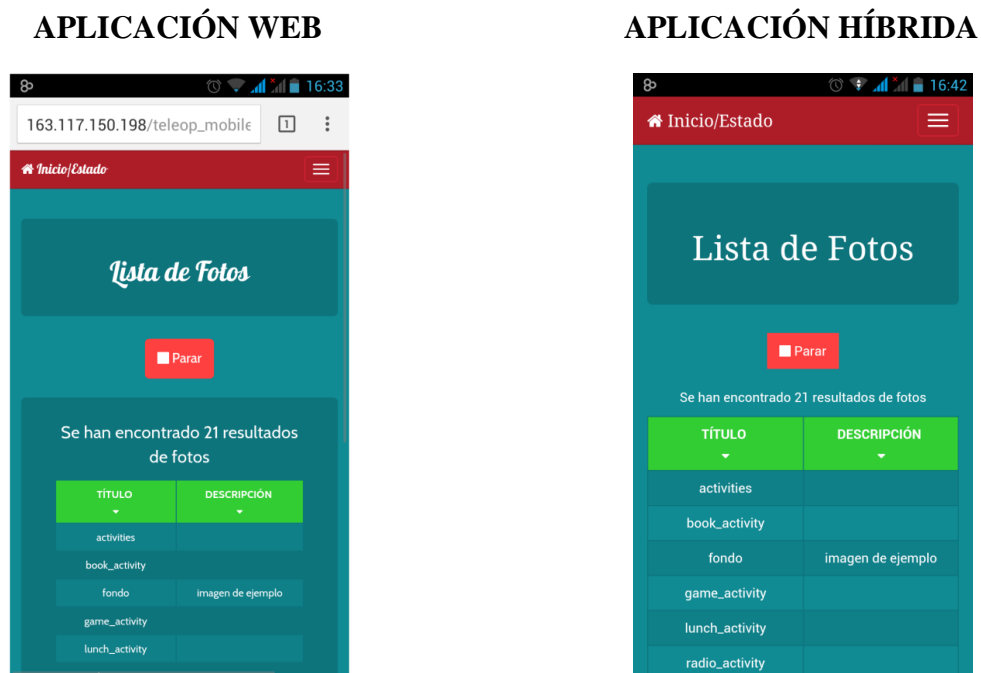


Figura B.9 - Página de la funcionalidad “música”

### B.7.2 Fotos

Al seleccionar la funcionalidad fotos, aparecerá la página con la lista de fotos disponibles (figura B.10).

Para que el robot deje de mostrar la foto pulse el botón rojo “Parar”.

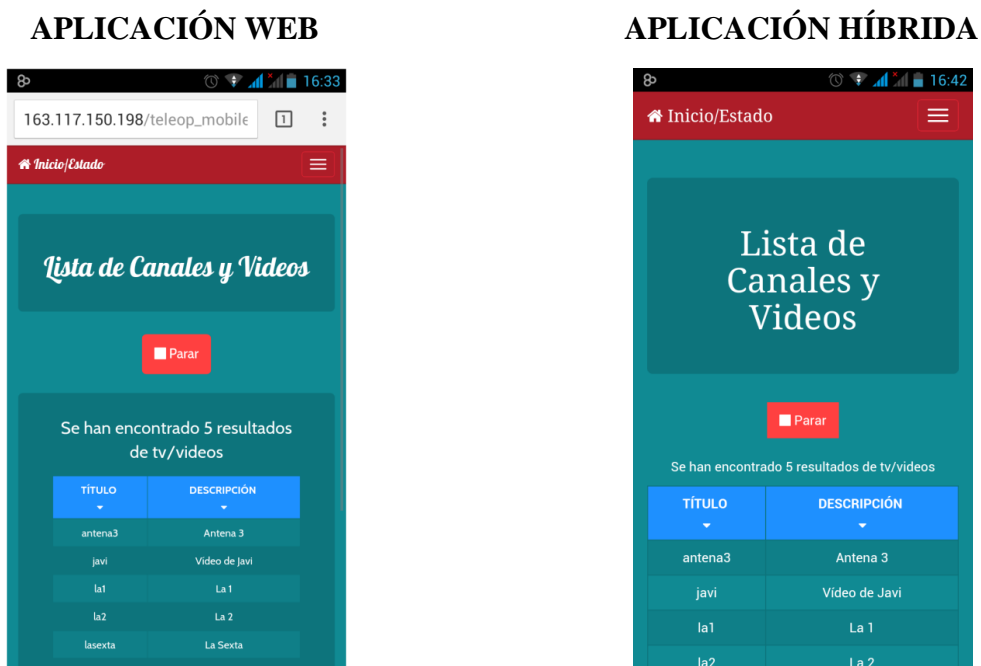


*Figura B.10 - Página de la funcionalidad “fotos”*

### B.7.3 Tv/Videos

Al seleccionar la funcionalidad tv/videos, aparecerá la página con la lista de canales y videos disponibles (figura B.11).

Para que el robot deje de reproducir el video o canal pulse el botón rojo “Parar”.

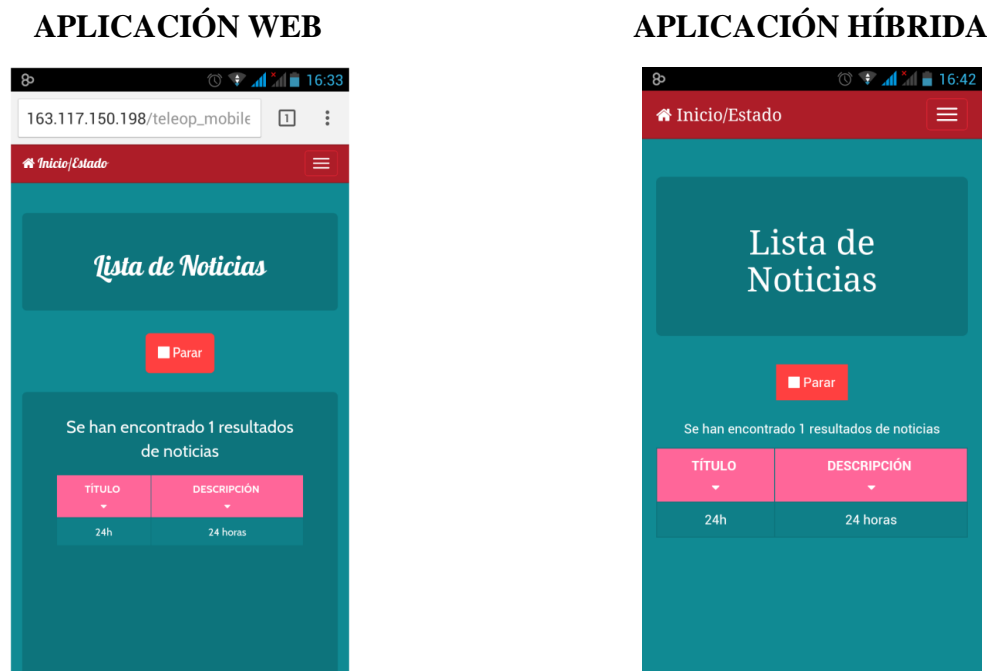


*Figura B.11 - Página de la funcionalidad “tv/videos”*

### B.7.4 Noticias

Al seleccionar la funcionalidad noticias, aparecerá la página con la lista de canales de noticias disponibles (figura B.12).

Para que el robot deje de reproducir el canal de noticias pulse el botón rojo “Parar”.

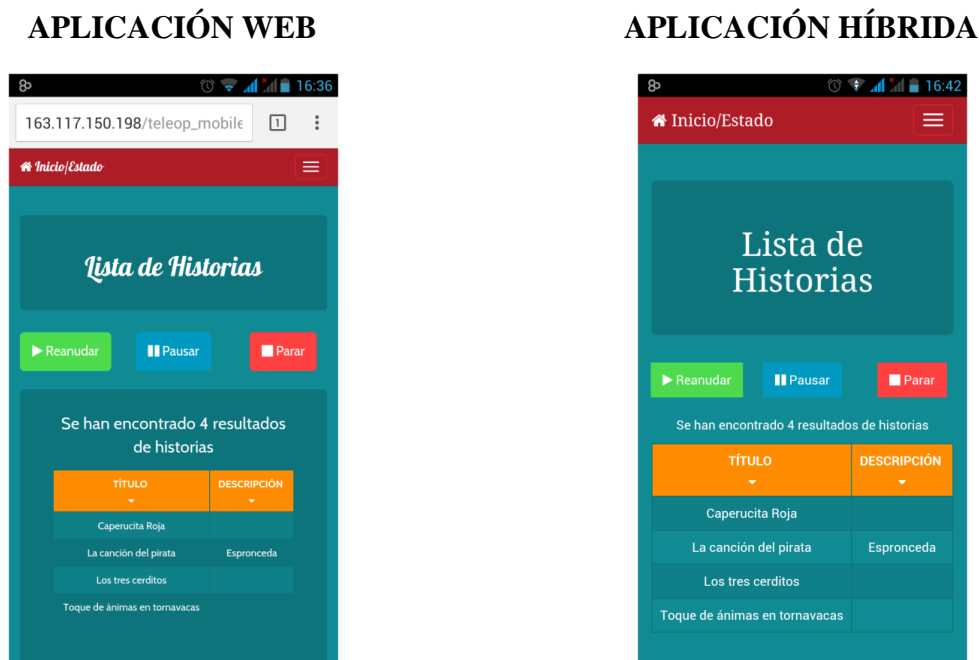


*Figura B.12 - Página de la funcionalidad “noticias”*

### B.7.5 Historias

Al seleccionar la funcionalidad historias, aparecerá la página con la lista de historias disponibles (figura B.13).

Para que el robot haga una pausa cuando está contando la historia pulse el botón azul “Pausar”. Si quiere volver a reanudar la historia desde el instante en el que se pausó, pulse el botón verde “Reanudar”. Para que deje de contar la historia pulse el botón rojo “Parar”.

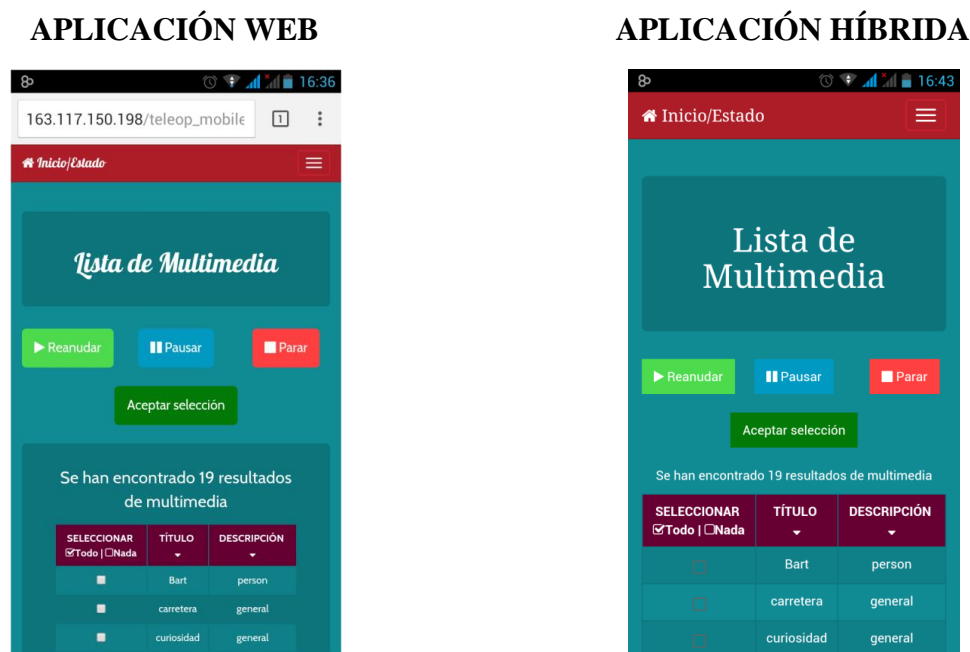


*Figura B.13 - Página de la funcionalidad “historias”*

## B.7.6 Multimedia

Al seleccionar la funcionalidad multimedia, aparecerá la página con la lista de etiquetas disponibles (figura B.14).

Para que el robot haga una pausa cuando está reproduciendo los contenidos multimedia pulse el botón azul “Pausar”. Si quiere volver a reanudar la tarea desde el instante en el que se pausó, pulse el botón verde “Reanudar”. Para que deje de reproducir la tarea pulse el botón rojo “Parar”.



*Figura B.14 - Página de la funcionalidad “multimedia”*

## B.8 Seleccionar tarea

Para que el robot realice una tarea se deben seguir los siguientes pasos:

- 1- Seleccionar la funcionalidad correspondiente: música, fotos, tv/videos, noticias, historias o multimedia.
- 2- Seleccionar la tarea deseada de la lista de tareas disponibles de la funcionalidad. Para la funcionalidad multimedia, seleccionar las etiquetas deseadas y después pulsar el botón “Aceptar selección”.
- 3- Pulsar “Aceptar” en el cuadro de confirmación

Cuando el robot comience a realizar la tarea, se mostrará en la aplicación un cuadro de información con la tarea que se ha seleccionado y se está realizando.

En los subapartados B.8.1 y B.8.2 se muestra un ejemplo de los pasos anteriores:

### B.8.1 Seleccionar canción

El cuadro de confirmación que aparece al seleccionar una tarea es el que se muestra en la figura B.15:

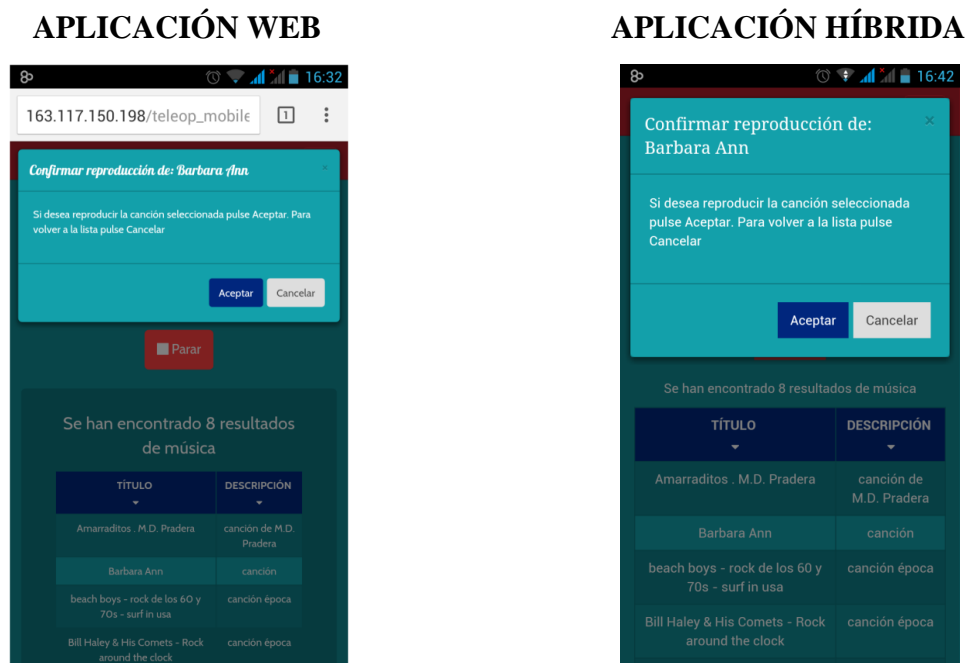


Figura B.15 - Cuadro de confirmación

Al aceptar la realización de la tarea, se muestra el cuadro de información de la figura B.16:



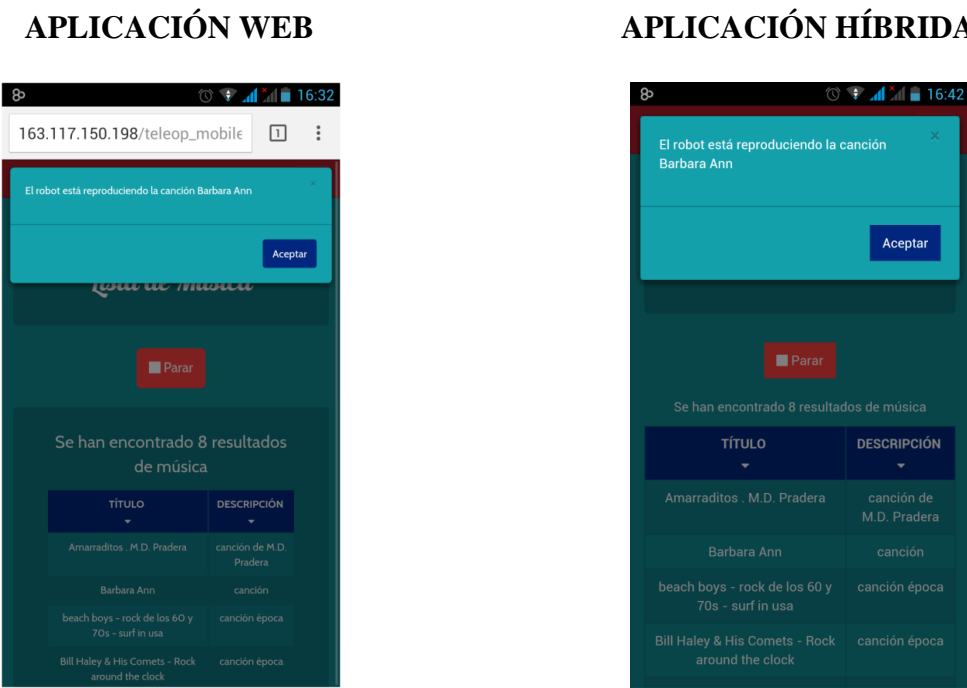


Figura B.16 - Cuadro de información

Puede comprobar en la página principal que se ha actualizado el estado con la tarea solicitada (figura B.17):

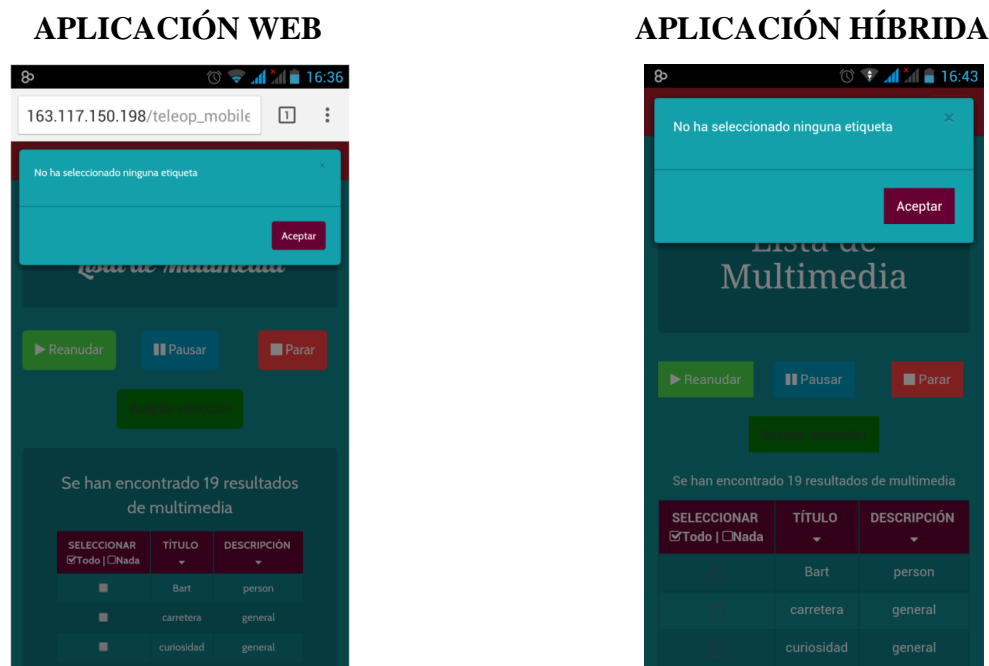


Figura B.17 - Actualización del estado

## B.8.2 Seleccionar etiquetas

La funcionalidad multimedia permite selección múltiple de etiquetas, es decir, el robot reproducirá el contenido multimedia de todas las etiquetas que seleccione el usuario.

Si no selecciona ninguna etiqueta y acepta la selección le aparecerá un cuadro de información indicando que no ha seleccionado nada como se muestra en la figura B.18. Al pulsar el botón “Aceptar” se volverá a la página de la funcionalidad multimedia para que vuelva a realizar una nueva selección de etiquetas.



*Figura B.18 - Cuadro de información al no seleccionar etiquetas*

En la columna “SELECCIONAR” se puede:

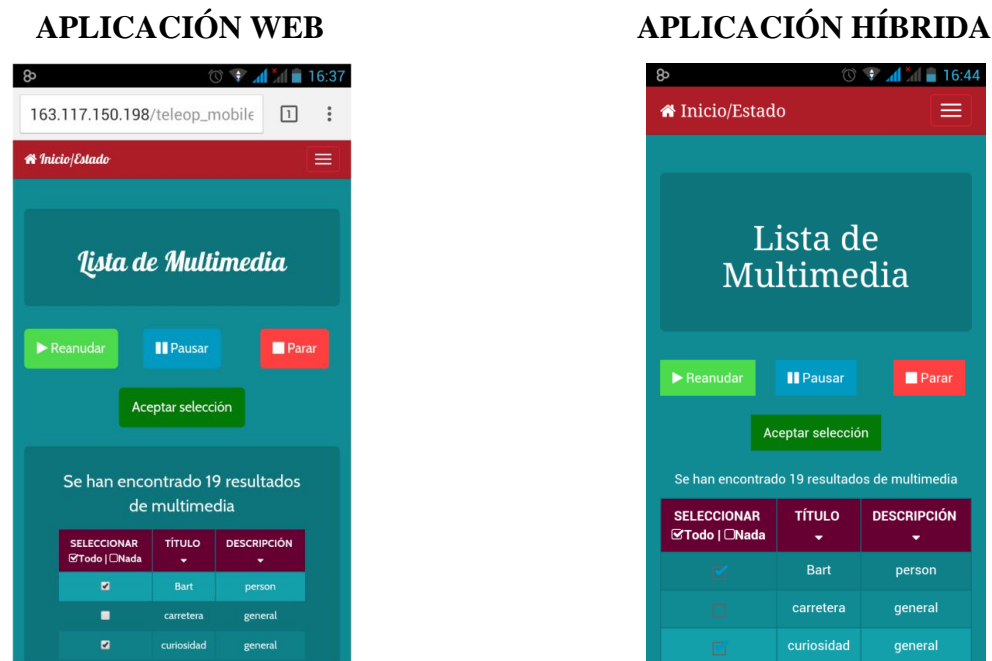
- Seleccionar todas las etiquetas pulsando “☒ Todo”, como muestra la figura B.19.
- Deseleccionar todas las etiquetas pulsando “☐ Nada”



*Figura B.19 - Selección de todas las etiquetas*

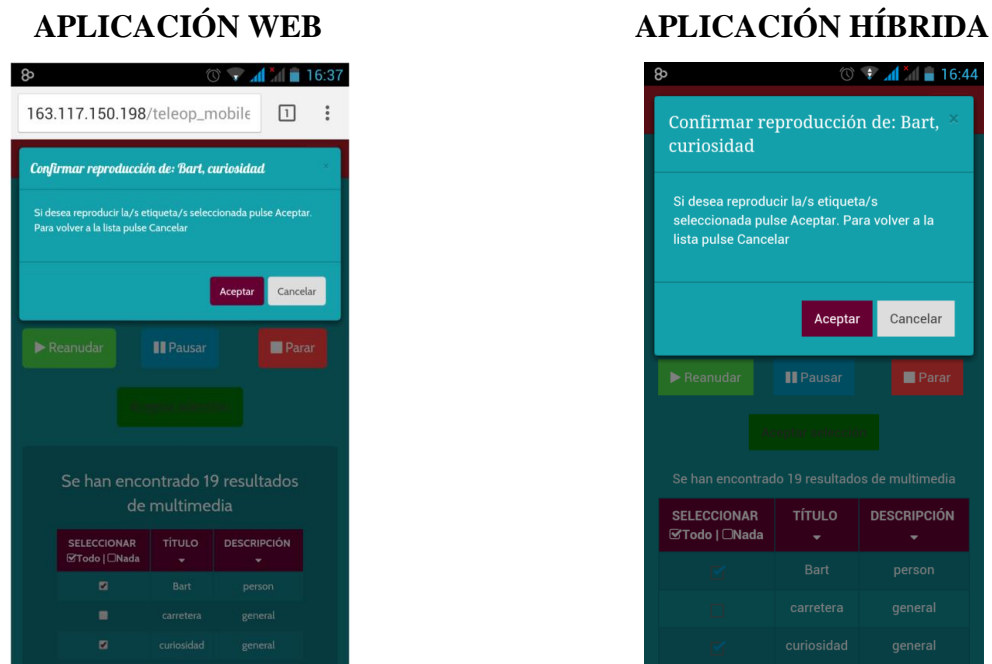
En este ejemplo,

- 1- Seleccionamos las etiquetas “Bart” y “Curiosidad” de la lista de etiquetas (figura B.20)
- 2- Aceptamos la selección pulsando “Aceptar selección”



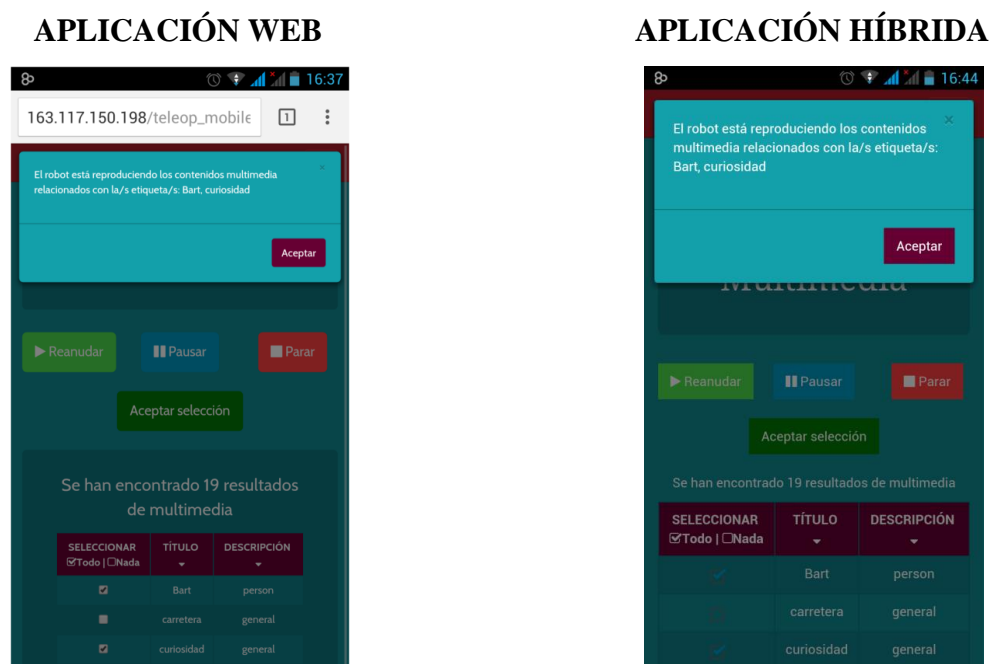
*Figura B.20 - Selección de algunas etiquetas*

- 3- Confirmamos la reproducción pulsando “Aceptar” en el cuadro de confirmación (figura B.21)



*Figura B.21 - Cuadro de confirmación de las etiquetas seleccionadas*

El robot reproducirá las etiquetas seleccionadas y recibiremos el mensaje de información con la tarea solicitada (figura B.22).



*Figura B.22 - Cuadro de información de las etiquetas en reproducción*

## B.9 Consultar ayuda

Para consultar la guía de ayuda que proporciona la aplicación (figura B.23):

- 1- Despliegue el menú.
- 2- Seleccione “Ayuda”



*Figura B.23 - Página de la guía de ayuda*

## B.10 Salir de la aplicación

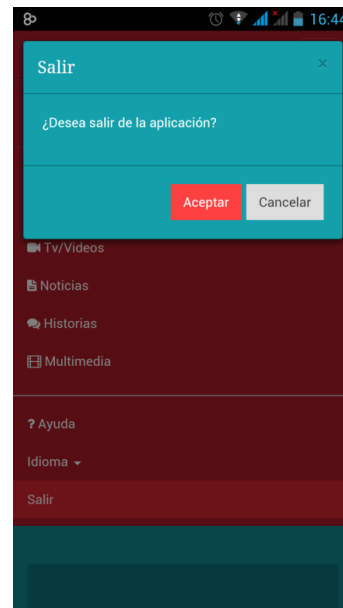
### APLICACIÓN WEB

Para salir de la aplicación hay que cerrar la pestaña del navegador y cerrar la aplicación navegador que hemos utilizado.

### APLICACIÓN HÍBRIDA

Para cerrar la aplicación “teleopMobile”:

- 1- Despliegue el menú.
- 2- Seleccione “Salir” (figura B.24).
- 3- Pulse “Aceptar” para salir de la aplicación.



*Figura B.24 - Cuadro de confirmación para salir de la aplicación*

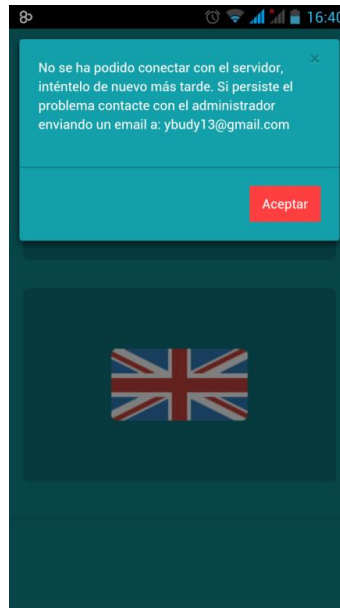
## **B.11 Mensajes de error de conexión**

Los mensajes de error de conexión son los que aparecen cuando no ha sido posible acceder al servidor web o a la base de datos. Estos mensajes sólo se muestran para el caso de la aplicación híbrida.

### **B.11.1 Error de conexión con el servidor**

Si el servidor web no se encuentra disponible le aparecerá el mensaje de la figura B.25. Al pulsar “Aceptar” la aplicación se cerrará.

## APLICACIÓN HÍBRIDA

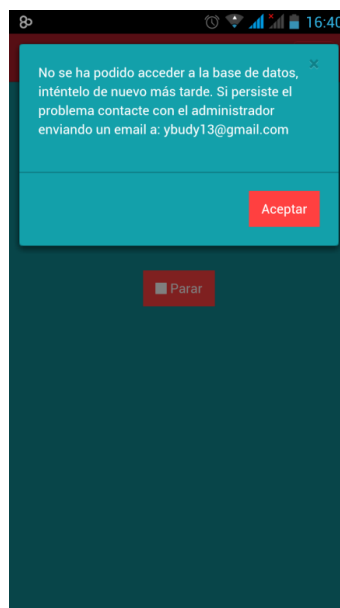


*Figura B.25 - Error de conexión con el servidor*

### B.11.2 Error de conexión con la base de datos

Si la base de datos no se encuentra disponible le aparecerá el mensaje de la figura B.26. Al pulsar “Aceptar” la aplicación se cerrará.

## APLICACIÓN HÍBRIDA



*Figura B.26 - Error de conexión con la base de datos*





# **Anexo C**

## **Tecnologías auxiliares**

En este anexo se describen las tecnologías utilizadas en el lado del servidor para facilitar el proceso de desarrollo de las aplicaciones, sin afectar la funcionalidad de éstas.

## C.1 PHPMyAdmin

Se trata de una herramienta de software libre escrita en PHP para administrar bases de datos MySQL mediante una página web intuitiva que permite al usuario realizar la mayoría de las operaciones MySQL tales como: buscar, crear, borrar, copiar, modificar bases de datos, tablas, vistas, campos, administrar usuarios y privilegios, importar/exportar datos, etc.

Se ha utilizado esta herramienta debido a que dispone de una gran documentación [41], de varios canales de soporte para obtener ayuda del equipo de PHPMyAdmin y está traducida a 72 idiomas. Es un proyecto de larga duración, estable y flexible, que ha ganado diversos premios.

Para utilizar PHPMyAdmin se debe introducir en la barra de direcciones del navegador web del equipo en el que se encuentre instalado Apache (servidor Web), la siguiente URL: *http://localhost/phpmyadmin*

## C.2 Roslaunch: ficheros de lanzamiento

ROS proporciona una herramienta que permite modificar los parámetros en el servidor de parámetros e iniciar diversos nodos ROS, incluyendo *roscore*, de una sola vez. Para ello son necesarios uno o más ficheros de lanzamiento, con extensión *.launch*, en los cuales se especifican los parámetros a modificar y los nodos que se quieren iniciar.

La diferencia entre *roslaunch* y *roslun* es que éste inicia un solo nodo y muestra la salida estándar por pantalla, mientras que *roslaunch* puede iniciar muchos nodos a la vez y la salida estándar es redirigida a un fichero de registro con extensión *.log*. Debido a que los nodos son iniciados aproximadamente al mismo tiempo, no se puede determinar el orden en el cual los nodos se van a iniciar.

Los ficheros de lanzamiento son documentos XML que deben tener un elemento raíz definido por el par de etiquetas *<launch>*, *</launch>*. Dentro de estas etiquetas es donde se van a definir el resto de elementos que se describen a continuación.

### ▪ Incluir otros ficheros de lanzamiento

Mediante el elemento *include*, un fichero de lanzamiento puede incluir el contenido de otros ficheros de lanzamiento, es decir, incluir todos sus nodos y parámetros.

```
<include file="path-to-launch-file" />
```

En lugar de introducir la ruta completa del fichero que se desea incluir, se puede sustituir por elementos que permiten buscar el paquete:

```
<include file="$(find package-name)/launch-file-name" />
```

### ▪ Espacio de nombres (*namespace*)

ROS soporta nombres relativos, de modo que permite utilizar el concepto de un espacio de nombres determinado. Utilizando este atributo es posible crear simulaciones independientes definiendo un espacio de nombres distinto para cada una.

```
ns="namespace"
```

### ▪ Grupos

Una manera conveniente de organizar los nodos en un fichero de lanzamiento largo es mediante el elemento *group*. Este elemento puede tener dos propósitos:

- Los nodos de dicho grupo tengan el mismo espacio de nombres

```
<group ns="namespace" />
...
</group>
```

- Permite condicionalmente activar o desactivar los nodos.

```
<group if="0-or-1" />
...
</group>
```

```
<group unless="1-or-0" />
...
</group>
```

### ▪ Nodos

Los nodos a iniciar se definen entre las etiquetas *node* y requieren al menos tres atributos: *pkg*, *type* y *name*.

```
<node
  pkg="package-name"
  type="executable-name"
  name="node-name"
/>
```

Los atributos *pkg* y *type* especifican el nombre del paquete y el nombre del ejecutable que identifican qué programa ROS se debe ejecutar para iniciar el nodo. Estos atributos son los mismos que los dos argumentos del comando *roslaunch*.

El atributo *name* asigna un nombre al nodo. Sobrescribe cualquier nombre que el nodo se asignaría en la llamada a *ros::init*.

Se puede utilizar el atributo *output="screen"* para mostrar la salida estándar por pantalla en lugar de almacenarla en un fichero de registro.

Mediante el atributo *required="true"* al finalizar un nodo, *roslaunch* termina con el resto de nodos activos y sale de sí mismo.

### ▪ Argumentos

Para facilitar la ejecución de los ficheros de lanzamiento, *roslaunch* soporta argumentos (*args*) cuya función es similar a las variables locales en un programa ejecutable.

Se declara la existencia del argumento mediante el elemento *arg*:

```
<arg name="arg-name" />
```

El argumento utilizado en el fichero de lanzamiento debe tener un valor asignado. Dicho valor se puede proporcionar por línea de comandos al ejecutar *roslaunch*.

```
roslaunch package-name launch-file-name arg-name:=arg-value
```

Una vez que el argumento está declarado y tiene un valor asignado, se puede usar dicho valor con:

```
$(arg arg-name)
```

### ▪ Fichero *activity\_persistence.launch*

Teniendo en cuenta los elementos y características de los ficheros de lanzamiento descritas, se ha realizado uno denominado “*activity\_persistence.launch*” (figura C.1) que inicia *roscore* si éste no ha sido iniciado previamente, incluye el fichero “*rosbridge\_websocket.launch*” e inicia el nodo “*activity\_persistence\_module*” que contiene el código de los servicios “*current\_activity*” y “*request\_activity*” y del tema “*activity\_finished*”.

```
<launch>
  <arg name="robot"/>
  <include file="$(find
rosbridge_server)/launch/rosbridge_websocket.launch" />
  <group ns="$(arg robot)">
    <node pkg="activity_persistence"
type="activity_persistence"
      name="activity_persistence_module" output="screen"
required="false" />
  </group>
</launch>
```

Figura C.1 - Código *activity\_persistence.launch*

Para ejecutar el fichero “*activity\_persistence.launch*”, se debe escribir en la terminal de Linux el código de la figura C.2:

```
roslaunch activity_persistence activity_persistence.launch robot:=alz
```

Figura C.2 - Código para ejecutar *activity\_persistence.launch*

## Acrónimos

ADT: *Android Developer Tools* – Herramientas de Desarrollo Android

AIMC: Asociación para la Investigación de Medios de Comunicación

AJAX: *Asynchronous JavaScript and XML* – JavaScript Asíncrono y XML

API: *Application Programming Interface* – Interfaz de Programación de Aplicaciones

ASCII: *American Standard Code for Information Interchange*

ASF: *Apache Software Foundation*

BBDD: Base de Datos

BSD: *Berkeley Source Distribution* - Distribución Fuente de Berkeley

CSS: *Cascading Style Sheets* – Hojas de Estilo en Cascada

DOM: *Document Object Model*

EA: Enfermedad de Alzheimer

FAE: Fundación Alzheimer España

GPL: *GNU General Public Licence* – Licencia Pública General de GNU

HTML: *Hyper Text Markup Language* – Lenguaje para Marcado de HiperTexto

HTTP: *Hypertext Transfer Protocol* – Protocolo de Transferencia de Hipertexto

IDC: *International Data Corporation* - Corporación de Datos Internacional

INe: Instituto Nacional de estadística

JDK: *Java Development Kit* - Kit de Desarrollo de Java

JSON: *JavaScript Object Notation* – Notación de Objetos de JavaScript

LPGL: *GNU Lesser General Public License* - Licencia Publica General de GNU

LTS: *Long Term Support*

MIT: *Massachusetts Institute of Technology* – Instituto Tecnológico de Massachusetts

OSI: *Open System Interconnection*

PDA: *Personal Digital Assistant* – Asistente Digital Personal  
PHP: *PHP Hypertext Preprocessor*  
RF: Requisito Funcional  
RI: Requisito de Información  
RIM: *Research In Motion* – Investigación en Movimiento  
RNF: Requisito No Funcional  
ROS: *Robot Operating System* – Sistema Operativo Robótico  
SDK: *Software Development Kit* – Kit de Desarrollo de Software  
SGML: *Standard Generalized Markup Language*  
SO: Sistema Operativo  
SQL: *Structure Query Language* – Lenguaje de Consulta Estructurado  
SSQLS: *Specialized SQL Structures* – Estructuras especializadas SQL  
TCP/IP: *Transmission Control Protocol/Internet Protocol*  
UC3M: Universidad Carlos III de Madrid  
URL: *Uniform Resource Locator* – Localizador Uniforme de Recursos  
W3C: *World Wide Web Consortium*  
XAMPP: X (para cualquier sistema operativo), Apache, MySQL, PHP, Perl  
XHTML: *eXtensible HyperText Markup Language*  
XML: *eXtensible Markup Language* - Lenguaje de Marcas eXtensible  
XSLT: *eXtensible Stylesheet Language Transformations*

## Referencias

- [1] INe (Instituto Nacional de Estadística), “*Notas de Prensa: Proyección de la Población de España a Corto Plazo 2013–2023*”, 22 de Noviembre de 2013. Sitio Web: [www.ine.es/prensa/prensa.htm](http://www.ine.es/prensa/prensa.htm)
- [2] Pwc y Lilly, “*Estado del arte de la enfermedad de Alzheimer en España*”, Madrid, Tech. Rep., 2013.
- [3] Jordi Peña-Casanova, “*Activemos la mente. Enfermedad de Alzheimer. Del diagnóstico a la terapia: conceptos y hechos*”. Fundación la Caixa.
- [4] Miguel A. Salichs, Irene P. Encinar, Esther Salichs, Alvaro Castro-Gonzalez, María Malfaz, “*Estudio de escenarios de uso para un robot social asistencial para enfermos de Alzheimer*”. RoboticsLab UC3M 2014.
- [5] Alexander Cerón Correa, “*Sistemas robóticos teleoperados*”, Ciencia e Ingeniería Neogranadina, 2005
- [6] Sitio Web de ROS. <http://www.ros.org/>
- [7] AIMC (Asociación para la Investigación de Medios de Comunicación), “*Marco general de los medios en España 2014*”, Madrid, 2014.
- [8] Sitio Web de IDC (*International Data Corporation*). <http://www.idc.com/>
- [9] Sitio Web de documentación de *rojava*. <http://wiki.ros.org/rojava>
- [10] Sitio Web de documentación de *rosbridge* [http://wiki.ros.org/rosbridge\\_suite](http://wiki.ros.org/rosbridge_suite)
- [11] T. Bray, “*The JavaScript Object Notation (JSON) Data Interchange Format*”, RFC 7159, RFC Editor, March 2014.
- [12] Sitio Web de documentación de *roslibjs* <http://wiki.ros.org/roslibjs>
- [13] Sitio Web de Twitter Bootstrap 3. <http://getbootstrap.com/>
- [14] Sitio Web de Github <https://github.com/>
- [15] Sitio Web de temas gratuitos de Bootstrap. <http://bootswatch.com/>
- [16] Sitio Web de Font-Awesome <http://fontawesome.github.io/Font-Awesome/>
- [17] Sitio Web de Foundation. <http://foundation.zurb.com/>

- 
- [18] Sitio Web de Skeleton. <http://getskeleton.com/>
  - [19] Sitio Web de jQuery Mobile <http://jquerymobile.com/>
  - [20] Sitio Web de ThemeRoller <http://themeroller.jquerymobile.com/>
  - [21] Sitio Web de Phonegap. <http://phonegap.com/>
  - [22] Sitio Web de Sencha Touch <http://www.sencha.com/products/touch/>
  - [23] Sitio Web de World Wide Web Consortium (W3C) <http://www.w3.org/>
  - [24] Sitio Web de CSS <http://www.w3.org/TR/CSS>
  - [25] Sitio Web de jQuery <http://jquery.com/>
  - [26] Sitio Web de Apache 2.2 <http://httpd.apache.org/>
  - [27] Sitio Web de XAMPP <https://www.apachefriends.org/index.html>
  - [28] Sitio Web de documentación PHP <http://php.net/>
  - [29] Sitio Web de C++ <http://www.cplusplus.com/>
  - [30] Sitio Web de MySQL <http://www.mysql.com/>
  - [31] Sitio Web de documentación MySQL++ <http://tangentsoft.net/mysql++/doc/html/refman/index.html>
  - [32] R. Fielding and J. Reschke, “*HTTP/1.1 Message Syntax and Routing*”, RFC 7230, RFC Editor, June 2014.
  - [33] Sitio Web del protocolo Rosbridge. [https://github.com/RobotWebTools/rosbridge\\_suite/blob/groovy-devel/ROSBRIDGE\\_PROTOCOL.md](https://github.com/RobotWebTools/rosbridge_suite/blob/groovy-devel/ROSBRIDGE_PROTOCOL.md)
  - [34] Jesse James Garrett, “*Ajax: A New Approach to Web Applications*”, Adaptive path, 18 de febrero de 2005
  - [35] Sitio Web de documentación Phonegap para Android [http://docs.phonegap.com/en/edge/guide\\_platforms\\_android\\_index.md.html#Android%20Platform%20Guide](http://docs.phonegap.com/en/edge/guide_platforms_android_index.md.html#Android%20Platform%20Guide)
  - [36] Sitio Web de Android Studio <http://developer.android.com/sdk/index.html>
  - [37] Sitio Web de Eclipse <http://eclipse.org/>
  - [38] Sitio Web de documentación para instalar ADT <http://developer.android.com/sdk/installing/installing-adt.html>
  - [39] Sección III, Ministerio de empleo y seguridad social, BOE (Boletín Oficial del Estado) nº 35 del martes 10 de febrero de 2015
  - [40] Sección III, Ministerio de empleo y seguridad social, BOE (Boletín Oficial del Estado) nº 36 del martes 11 de febrero de 2015
  - [41] Sitio Web de PHPMyAdmin <http://www.phpmyadmin.net/>